

PROJECT REPORT
ON
“POLICE BHARTI”
FOR
PERFECT COMPUTERS INFOTECH

Submitted By
Kusekar Sagar Sanjay
Shirdhankar Nikhil Mangesh

Master of Computer Application
Savitribai Phule Pune University
Vidya Pratishthan's Institute of Information Technology (VIIT),
(ACCREDITED BY NAAC WITH “A” GRADE)
Vidya Nagari, Baramati,
Dist. Pune 413 133

(2018-2021)

CERTIFICATE

This is to certify that, Mr. Shirdhankar Nikhil Mangesh of MCA Course (Track-I: Software and Application Development), SEM- V and VI has completed his Project Work Titled "**Police Bharti**", as a part of curriculum, during the Academic Year 2020-21.

University Seat No:

Dr. Mayank Kothawade

Head of Department

Dr. Sateeshchandra Joshi

In-charge Director

External Examiner

Sign of Examiners:

1. _____

2. _____

Date of Examination:



CERTIFICATE

This is to certify that, Mr Kusekar Sagar Sanjay of MCA Course (Track-I: Software and Application Development), SEM- V and VI has completed his Project Work Titled "**Police Bharti**", as a part of curriculum, during the Academic Year 2020-21.

University Seat No:

Dr. Mayank Kothawade

Head of Department

Dr. Sateeshchandra Joshi

In-charge Director

External Examiner

Sign of Examiners:

1. _____

2. _____

Date of Examination:



GUIDE CERTIFICATE

This is to certify that, Mr. Shirdhankar Nikhil Mangesh of MCA Course SEM- V and VI has successfully completed his Project Work Titled "**Police Bharti**", under my guidance during the Academic Year 2020-21.

Dr. Mayank Kothawade

Project Guide



GUIDE CERTIFICATE

This is to certify that, Mr. Kusekar Sagar Sanjay of MCA Course SEM- V and VI has successfully completed his Project Work Titled “**Police Bharti**”, under my guidance during the Academic Year 2020-21.

Dr. Mayank Kothawade

Project Guide



DECLARATION

We hereby declare that, the project entitled "**Police Bharti**" for "**Perfect Computer Infotech**", have completed and written by us. Current project has not previously formed the basis for the award of any Degree or Diploma or other similar title of this or any other University or examining body.

Place: Baramati

Date:

Shirdhankar Nikhil Mangesh

Kusekar Sagar Sanjay

Signature

Signature



Empowering Tomorrow.

Completion Certificate

This is to certify that **Mr. Shirdhankar Nikhil Mangesh** of **Vidya Pratishthan's Institute of Information Technology (VIIT)**, Baramati has successfully completed their Third Year project in partial fulfillment of MCA degree at PCT InfoTech.

They have undertaken and completed project titled **Police Bharti System** using ASP.NET with C# Database MySQL. The project commence from 1 August 2020 to till date. They have worked sincerely to our complete satisfaction.

They are hardworking and ready to take challenges, I wish them rewarding career ahead.



A handwritten signature in blue ink, followed by the text "For PERFECT COMPUTERS" and "Proprietor" in a blue sans-serif font.

Vasim G. Aowte
Proprietor
(MCA, MCP)



Empowering Tomorrow.

Completion Certificate

This is to certify that **Mr. Kusekar Sagar Sanjay** of **Vidya Pratishthan's Institute of Information Technology (VIIT)**, Baramati has successfully completed their Third Year project in partial fulfillment of MCA degree at PCT InfoTech.

They have undertaken and completed project titled **Police Bharti System** using ASP.NET with C# Database MySQL. The project commence from 1 August 2020 to till date. They have worked sincerely to our complete satisfaction.

They are hardworking and ready to take challenges, I wish them rewarding career ahead.



A handwritten signature in blue ink, followed by the text "For PERFECT COMPUTERS" and "Proprietor" written in blue ink below it.

Vasim G. Aowte
Proprietor
(MCA, MCP)

ACKNOWLEDGEMENT

We take pleasure in presenting our project work. We are thankful to our project guide **Dr. Mayank Kothawade** of VIIT, **Vasim G. Aowte** of **Perfect Computers Infotech**, who keeps monitoring my / our project continuously to complete it in time as per the expectations of the course.

“Ability is of little account without opportunity”, I / we wish to thank our In-charge Director **Dr. Sateeshchandra Joshi**, who gave me / us a very bright learning opportunity, during my / our course to prove my / our ability and skills.

I / We would also like to thank **Dr. Mayank Kothawade** (Head of Department) for providing all the necessary facilities at college during the development of this project.

Last but not least I / We would like to express my / our heartfelt gratitude towards staff members of VIIT, my / our colleagues and friends for their moral and technical support throughout the duration of the project.

My / Our sincere thanks to all

1. **Kusekar Sagar Sanjay**
 2. **Shirdhankar Nikhil Mangesh**
-

TABLE OF CONTENTS

Chapter No.	Topic Name	Page No.
1	INTRODUCTION	1-2
	1.1 Company Profile	1
	1.2 Introduction of project	1
	1.3 Existing system & need of the system	1
	1.4 Limitations of Existing system	2
2	PROPOSED SYSTEM	3-23
	2.1 Problem Statement/s	3
	2.2 Product Position Statement/s	5
	2.3 Product Overview	5
	2.4 Technologies Used	5
	2.5 Summary of Capabilities	11
	2.6 Assumptions & Dependencies	12
	2.7 Objectives of the Proposed System.	12
	2.8 Functional requirements / Salient Features	13
	2.9 Non functional requirement/s	16
	2.10 Stake holder Summary	16
	2.11 User Summary	17
	2.12 Scope of the System	19
	2.13 Module Specification	20
2.14 Operational Environments	23	
3	REQUIREMENT DETERMINATION & ANALYSIS	24-26
	3.1 Fact Finding Methods	24
	3.2 Feasibility Study	25
4	SYSTEM ANALYSIS AND DESIGN	27-88
	4.1 Use Case Diagram	27
	4.2 Sequence Diagram	36
	4.3 Activity Diagram	45

Chapter No.	Topic Name		Page No.
	4.4	Class Diagram	54
	4.5	Object Diagram	55
	4.6	Deployment Diagram	56
	4.7	User Interface Design (Screens etc.)	57
	4.8	Table specifications (In case back end is a database)	67
5	USER MANUAL		89-91
	5.1	User Manual	89
	5.2	Operations Manual / Menu Explanation	89
	5.3	Program Specifications / Flow Charts	90
6	BENEFIT, LIMITATIONS AND CONCLUSION OF THE SYSTEM		92
	6.1	Benefits of the system	92
	6.2	Limitations of the system	92
	6.3	Conclusion of the system	
7	PROPOSED ENHANCEMENT		93
8	SELF-LEARNING FROM PROJECT (OVERALL EXPERIENCE)		94
BIBLIOGRAPHY			95
	I	Books / Journals	95
	II	Links of websites referred	95
ANNEXURE – I User Interface Screens			
ANNEXURE – II Output Reports with data (if any)			
ANNEXURE – III Sample Program Code / Project Demo (which will prove sufficient development is done by the student)			

Chapter I

INTRODUCTION

1.1 Company Profile :

PCinfotech is a private limited company which provides high quality services in software development at affordable costs. Our office is located in Ratnagiri, Maharashtra India.

We offer services like:

- Software Development
- Web Development

We use technologies such as .NET, Android, Java Script (angular, react), PHP. We also provide SEO, Social media advertising & Marketing under Digital Marketing Services.

PCinfotech was started in 2004 and has established and maintained good relations with their clients. The head of the company has working experience of 25 years.

1.2 Introduction of Project :

This is a web based portal application developed for police bharti. This portal provides easy to use interface and keeps track of scores of the candidates and makes it easy to schedule further events. The system will be used by highest authority of police recruitment. System makes the process easy by automating the process of importing the data from excel files.

1.3 Existing system & need of the system :

Existing system contains a windows based application. Windows based application is easy to use but it contains some flaws. These flaws maybe small but they have a huge impact on the system. The existing system faces the following problems:

- This windows based application has a local database and needs to be updated manually.
- As the windows based application has local database, it runs on local machine.

- The data of physical, ground and written tests which is verified by the respective authorities on paper sheets which are later added manually in the windows application.
- This verified data can be manipulated easily by someone in the process and it can't be tracked who manipulated the data.
- No ability to plan future events (such as dates of written tests, medicals etc.)
- The system cannot be used at different locations.
- The proposed system is web based application which maintains a centralized database of all exams / events taken by authorities.
- The informative data / reports can be stored in centralized database which can be maintained by the system and which gives good security to information.

1.4 Limitations of Existing system :

- As the windows based application has local database, it runs on local machine.
- The data of physical, ground and written tests are submitted by the respective authorities on paper sheets which are later added manually in the windows application.
- This data can be manipulated easily by someone in the process and it can't be tracked who manipulated the data.

Chapter II

PROPOSED SYSTEM

2.1 Problem Statements:

Problem No. 1

The Problem of	Manage users
The impact of which is	Problems in managing users.
A successful solution would	Proposed system can provide the easy way to manage the users.

Problem No. 2

The Problem of	Planning Event dates
The impact of which is	
A successful solution would	Proposed system can provide the easy way for planning the dates

Problem No. 3

The Problem of	Set the criteria for an event / exam.
The impact of which is	
A successful solution would	Proposed system can provide the easy way to set criteria and its points.

Problem No. 4

The Problem of	Registration of users under PSI
The impact of which is	For physical and ground test
A successful solution would	Proposed system can provide the easy way to register the members for the tests.

Problem No. 5

The Problem of	Registration of users under DSP
The impact of which is	For written test
A successful solution would	Proposed system can provide the easy way to register the members for the exam

Problem No. 6

The Problem of	Generates sheets based on report
The impact of which is	Excel sheet
A successful solution would	Proposed system can provide the easy way to generate the excel sheet based on report

Problem No. 7

The Problem of	Comment while updating the records
The impact of which is	Manipulation of data
A successful solution would	Proposed system can provide the easy way to add a comment while updating any record

Problem No. 8

The Problem of	Role based login
The impact of which is	Security of data
A successful solution would	Proposed system can provide the easy way to log in based on their role

2.2 Product Position Statement:

For	Police bharti Authorities
Who	Police bharti Authorities
The Police bharti	is a web based portal system
That	
Unlike	Windows based application
Our product	Provides features like user maintenance, user access control, report generators (Excel sheet) etc. to the system.

2.3 Product Overview:

“Police Bharti” is web based portal application developed for police requirements. The Admin can put all kind of information about the recruitment process and the dates for the exam / events.

- This web based system also used for manage and schedule the dates.
- This system provides the best security for logins based on their roles.
- Easy and simple User interface
- The proposed system is web based portal application which maintains a centralized database of all exams / events taken by authorities.

2.4 Technologies Used:

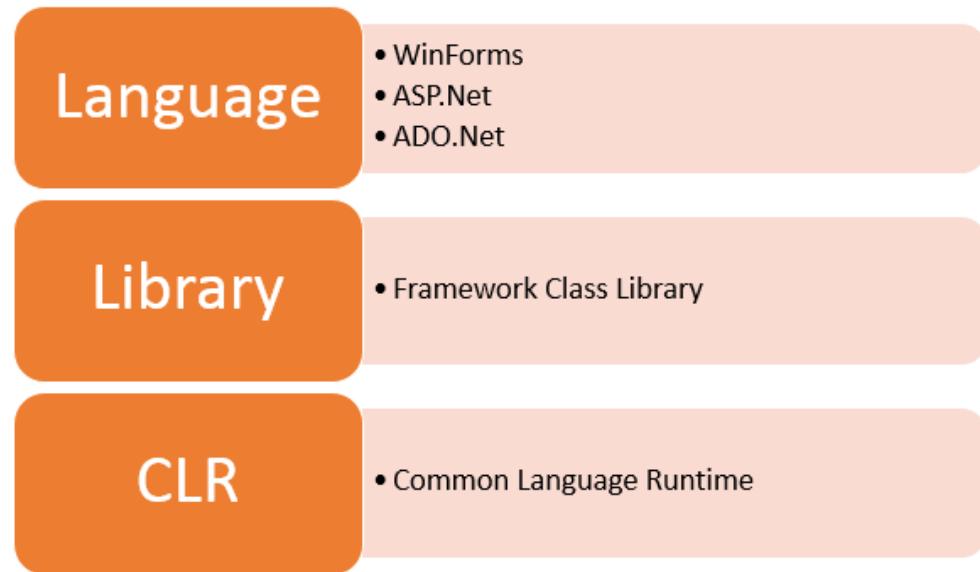
.NET FRAMEWORK

.Net Framework is a software development platform developed by Microsoft for building and running Windows applications. The .Net framework consists of developer tools, programming languages, and libraries to build desktop and web applications. It is also used to build websites, web services, and games.

The .Net framework was meant to create applications, which would run on the Windows Platform. The first version of the .Net framework was released in the year 2002. The version was called .Net framework 1.0. The Microsoft .Net framework has come a long way since then, and the current version is .Net Framework 4.7.2. The Microsoft .Net framework can be used to create both - **Form-based** and **Web-based** applications. Web services can also be developed using the .Net framework.

.Net Framework Architecture

.Net Framework Architecture is a programming model for the .Net platform that provides an execution environment and integration with various programming languages for simple development and deployment of various Windows and desktop applications. It consists of class libraries and reusable components.



.NET Components

The architecture of .Net framework is based on the following key components;

1. Common Language Runtime

The "Common Language Infrastructure" or CLI is a platform in .Net architecture on which the .Net programs are executed.

The CLI has the following key features:

- Exception Handling - Exceptions are errors which occur when the application is executed.

Examples of exceptions are:

- If an application tries to open a file on the local machine, but the file is not present.
- If the application tries to fetch some records from a database, but the connection to the database is not valid.
- Garbage Collection - Garbage collection is the process of removing unwanted resources when they are no longer required.

Examples of garbage collection are

- A File handle which is no longer required. If the application has finished all operations on a file, then the file handle may no longer be required.
- The database connection is no longer required. If the application has finished all operations on a database, then the database connection may no longer be required.

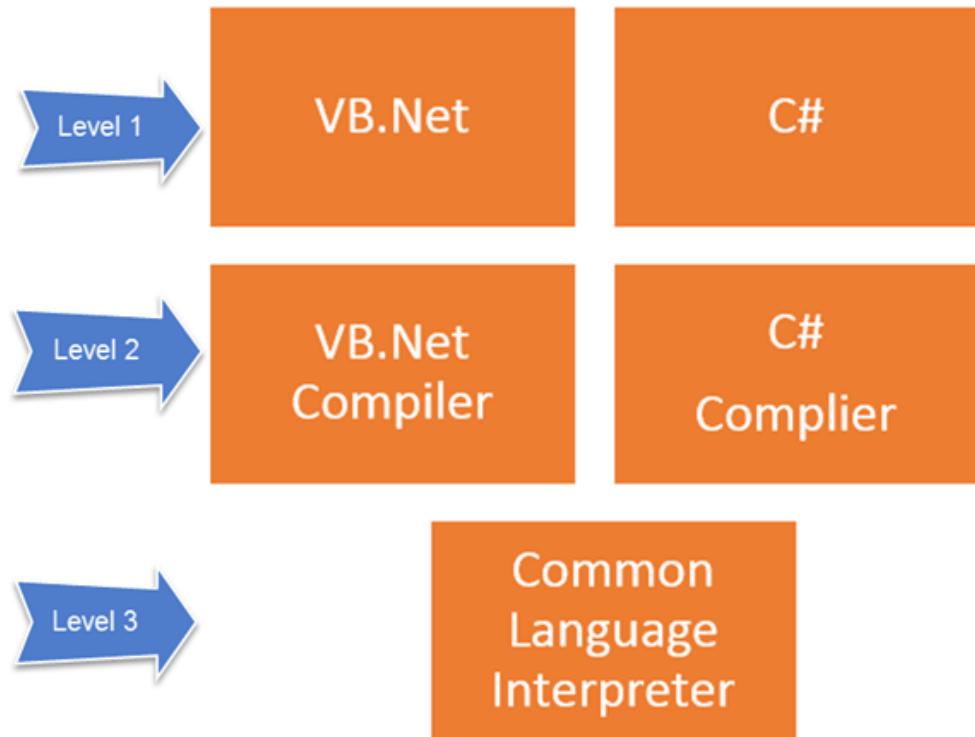
- Working with Various programming languages –

As noted in an earlier section, a developer can develop an application in a variety of .Net programming languages.

1.Language - The first level is the programming language itself, the most common ones are VB.Net and C#.

2.Compiler – There is a compiler which will be separate for each programming language. So underlying the VB.Net language, there will be a separate VB.Net compiler. Similarly, for C#, you will have another compiler.

3.Common Language Interpreter – This is the final layer in .Net which would be used to run a .net program developed in any programming language. So the subsequent compiler will send the program to the CLI layer to run the .Net application.



2. Class Library

The .NET Framework includes a set of standard class libraries. A class library is a collection of methods and functions that can be used for the core purpose. For example, there is a class library with methods to handle all file-level operations. So there is a method which can be used to read the text from a file. Similarly, there is a method to write text to a file.

Most of the methods are split into either the System.* or Microsoft.* namespaces. (The asterisk * just means a reference to all of the methods that fall under the System or Microsoft namespace)

A namespace is a logical separation of methods. We will learn these namespaces more in detail in the subsequent chapters.

3. Languages

The types of applications that can be built in the .Net framework is classified broadly into the following categories.

- Win Forms – This is used for developing Forms-based applications, which would run on an end user machine. Notepad is an example of a client-based application.
- ASP.Net – This is used for developing web-based applications, which are made to run on any browser such as Internet Explorer, Chrome or Firefox.
 - The Web application would be processed on a server, which would have Internet Information Services Installed.
 - Internet Information Services or IIS is a Microsoft component which is used to execute an Asp.Net application.
 - The result of the execution is then sent to the client machines, and the output is shown in the browser.
- ADO.Net – This technology is used to develop applications to interact with Databases such as Oracle or Microsoft SQL Server.
Microsoft always ensures that .Net frameworks are in compliance with all the supported Windows operating systems.

MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (<http://www.mysql.com/company/legal/licensing/>).

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

2.5 Summary of Capabilities:

“Police Bharti” is web based portal system developed for police requirements. The Admin can put all kind of information about the recruitment process and the dates for the exam / events.

Member Benefit	Supporting Features
Allows verified Member to do login	The system maintains the Member's details in the database for verification purpose.
Auto Import Excel sheet	System imports the provided Excel sheet into the database
Manage Users	Allows the administrator to manage the registered sub admins, he has the power to delete the specific user at any given time
Planning Event Dates	Dates of the events can be set by the district admin and City, Gramin Admins can pick a suitable date from that dates.
Criteria for Events	District admin can set the criteria for each event separately
Search specific candidate details	Search the details of a specific candidate from database, Any admin can check scores of that candidate at any given time
Generate Excel based reports	District admin can view reports of the candidates after the events take place. He

	can also generate excel reports if necessary.
View reports on each event separately	The system can show reports on each event separately or combined

2.6 Assumptions & Dependencies:

Assumptions:-

- 1.Member must have valid information, so that the admin can do the registration of the member.
- 2.Member could login with proper user name and password.

Dependencies:-

- 1.There should be a network connection.
- 2.Each member has their authentication.

2.7 Objectives of Proposed System:

- To develop the system that enables efficient management
- Providing facility to admin to create the multiple members.
- To show the result of every test exam at higher authority side.
- To add the list of candidates.
- To reduce time consumption.
- To provide more security for data
- To provide more security for login based on their role
- To approve the report of every test exam at admin side.

- Providing facility to admin to check every report at any time.
- To Review Scores of the candidates and update if necessary with remarks.

2.8 Functional Requirement's:

Number of modules:-

- 1.District Admin
- 2.City Admin
- 3.Gramin Admin
- 4.City Physical Admin
- 5.City Written Admin
- 6.City Medical Admin
- 7.Gramin Physical Admin
- 8.Gramin Written Admin
- 9.Gramin Medical Admin

1.District Admin:-

Administrator can add, delete and update member information. Administrator has all the functionality of members. Additionally administrator can deal with member's information and can deal with task related data.

2.City Admin:-

City Administrator is a police officer. He has the ability to register subordinates under him for Physical, written and medical test. Physical,

Written and Medical tests are conducted on the scheduled dates by City Admin.

3.Gramin Admin:-

Gramin Administrator is a police officer. He has the ability to register subordinates under him for Physical, Written and Medical tests. Physical, Written and Medical Tests are conducted on scheduled dates by Gramin Admin.

4.City Physical Admin:-

City Physical Administrator is a police officer whose registration is done by the City Administrator to conduct the Physical and Ground tests on the scheduled dates. The City Physical admin conducts the tests and adds the scores of the candidates in the database after verifying them.

5.City Written Admin:-

City Written Administrator is a police officer whose registration is done by the City Administrator to Conduct the Written test on the scheduled dates. City Written Administrator conducts the test and adds the scored of the candidates to the database after verifying them.

6.City Medical Admin:-

City Medical Administrator is a police officer whose registration is done by the City Administrator to conduct the Medical Tests on the scheduled dates. Medical tests are conducted at the Government Hospital and the results of the candidates are added to the database after verifying them. If the candidate fails in a certain test then he can reappear for medical test and the scores of that candidate are updated in the database.

7.Gramin Physical Admin:-

Gramin Physical Administrator is a police officer whose registration is done by the Gramin Administrator to conduct the Physical and Ground tests on the scheduled dates. The Gramin Physical admin conducts the tests and adds the scores of the candidates in the database after verifying them.

8.Gramin Written Admin:-

Gramin Written Administrator is a police officer whose registration is done by the Gramin Administrator to Conduct the Written test on the scheduled dates. Gramin Written Administrator conducts the test and adds the scored of the candidates to the database after verifying them.

9.Gramin Medical Admin:-

Gramin Medical Administrator is a police officer whose registration is done by the Gramin Administrator to conduct the Medical Tests on the scheduled dates. Medical tests are conducted at the Government Hospital and the results of the candidates are added to the database after verifying them. If the candidate fails in a certain test then he can reappear for medical test and the scores of that candidate are updated in the database.

2.9 Nonfunctional requirements:

The development of this new system contains all of the above functional requirements along with following activities:

- Fast Performance and high accuracy.
- Reliability and flexibility.
- The system is Member friendly and self-explanatory.
- The system is available 100% for the Member and is used 24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

2.10 Stake holder Summary:

These are the factors affecting the system:

1. Admin: Highest authority of police recruitment.
2. City Admin: - Member created by Admin for the authority of city police recruitment.
3. Gramin Admin: - Member created by Admin for the authority of gramin police recruitment.
4. City Physical Admin: - Member is created by City Admin to conduct physical and ground test and update score.
5. City Written Admin: - Member is created by City Admin to conduct written test and update the score.
6. City Medical Admin: - Member is created by City Admin to conduct medical and re-medical test and update remark.
7. Gramin Physical Admin: - Member is created by Gramin Admin to conduct physical and ground test and update score.

8. Gramin Written Admin: - Member is created by Gramin Admin to conduct written test and update the score.
9. Gramin Medical Admin: - Member is created by Gramin Admin to conduct medical and re-medical test and update remark.

Name	Represents	Role
	Highest authority of police recruitment	Controls all the activities of this application.

2.11 User Summary:

Name	Description	Stakeholder
Admin (District level Admin)	Highest authority of this system who has the authority of recruitment process	Admin is responsible to create the members for this whole recruitment process. Also he/she is responsible to assign the roles to the member. He/she has the authority to manage all members and its responsibilities.
Gramin Admin	Authority who supervises the procedure of conducting tests and	

	monitoring them from Gramin area.	
City Admin	Authority who supervises the procedure of conducting tests and monitoring them from City area.	
City Physical Admin	Authority to conduct the physical and ground test and send the overall report to the highest authority for further recruitment process.	
City Written Admin	Authority to conduct the written test exam and send the report to the highest authority for further recruitment process	
City Medical Admin	Authority who monitors the process of medical tests of the candidates and updates the scores accordingly.	
Gramin Physical Admin	Authority to conduct the physical and ground test and send the overall report to the highest authority for further recruitment process.	

Gramin Written Admin	Authority to conduct the written test exam and send the report to the highest authority for further recruitment process	
Gramin Medical Admin	Authority who monitors the process of medical tests of the candidates and updates the scores accordingly.	

2.12 Scope of the System:

- District Admin is able to plan events such as Physical, Ground, Written and Medical tests.
- District Admin is able to authenticate City and Gramin Admins for data entry and verification which are police officers.
- District Admin has the power to remove authentication of users if needed.
- District Admin is able to approve candidates according to the cast reservation.
- District Admin is able to see the verified results of physical and ground tests which are conducted by City and Gramin Admins.
- District Admin is able to see the verified results of written tests which are conducted by City and Gramin Admins.
- District Admin is able to see the verified results of medical tests which are conducted by City and Gramin Admins.
- District Admin is able to view the list of candidates with their combined aggregate marks.
- District Admin Provides the data of the candidates in excel sheets which gets stored in the database.
- District Admin is able to generate the final reports of the candidates and export them in the excel format.

- Authenticated City and District Admins are able to view the data of the candidates submitted by the Admin.
- City and District Admins are able to authenticate Physical, Written and Medical Admin each for ground, written and medical tests.
- Authenticated Physical Admins conduct the physical (weight & chest) and ground tests of the candidates submitted by the District Admin on scheduled date.
- Authenticated Physical Admins are able to add the scores of the candidates after conducting the tests.
- Physical Admins are able to view the scores and update if necessary with proper comments.
- Authenticated Physical Admins are able to view the reports of the candidates and generate excel file if necessary.
- Authenticated Written Admins are able to view the data submitted by the admin.
- Authenticated Written Admins carry out the written tests.
- Authenticated Written Admins are able to add the scored of the candidates after conducting the written tests.
- Written Test Admins are able to view the results of the candidates and update them if necessary with proper remarks.
- Written Test Admins are able to view the reports of the candidates and export them if necessary.

2.13 Module Specification:

Following are the members of “Police Bharti”

- 1.District Admin
 - 2.City Admin
 - 3.Gramin Admin
 - 4.City Physical Admin
 - 5.City Written Admin
-

- 6.City Medical Admin
- 7.Gramin Physical Admin
- 8.Gramin Written Admin
- 9.Gramin Medical Admin

Description of the Members:

District Admin:

Administrator can add, delete and update member information. Administrator has all the functionality of members. Additionally administrator can deal with member's information and can deal with task related data.

City Admin:

City Administrator is a police officer. He has the ability to register subordinates under him for Physical, written and medical test. Physical, Written and Medical tests are conducted on the scheduled dates by City Admin.

Gramin Admin:

Gramin Administrator is a police officer. He has the ability to register subordinates under him for Physical, Written and Medical tests. Physical, Written and Medical Tests are conducted on scheduled dates by Gramin Admin.

City Physical Admin:

City Physical Administrator is a police officer whose registration is done by the City Administrator to conduct the Physical and Ground tests on the scheduled dates. The City Physical admin conducts the tests and adds the scores of the candidates in the database after verifying them.

City Written Admin:

City Written Administrator is a police officer whose registration is done by the City Administrator to Conduct the Written test on the scheduled dates. City Written Administrator conducts the test and adds the scored of the candidates to the database after verifying them.

City Medical Admin:

City Medical Administrator is a police officer whose registration is done by the City Administrator to conduct the Medical Tests on the scheduled dates. Medical tests are conducted at the Government Hospital and the results of the candidates are added to the database after verifying them. If the candidate fails in a certain test then he can reappear for medical test and the scores of that candidate are updated in the database.

Gramin Physical Admin:

Gramin Physical Administrator is a police officer whose registration is done by the Gramin Administrator to conduct the Physical and Ground tests on the scheduled dates. The Gramin Physical admin conducts the tests and adds the scores of the candidates in the database after verifying them.

Gramin Written Admin:

Gramin Written Administrator is a police officer whose registration is done by the Gramin Administrator to Conduct the Written test on the scheduled dates. Gramin Written Administrator conducts the test and adds the scored of the candidates to the database after verifying them.

Gramin Medical Admin:

Gramin Medical Administrator is a police officer whose registration is done by the Gramin Administrator to conduct the Medical Tests on the scheduled dates. Medical tests are conducted at the Government Hospital and the results of the

candidates are added to the database after verifying them. If the candidate fails in a certain test then he can reappear for medical test and the scores of that candidate are updated in the database.

2.14 Operational Environments:

System Requirements

Server side	
Processor	1.6GHz or faster processor
RAM	4 GB
Hard Disk	20 GB of free space
Client side	
Processor	Intel core i3
RAM	4 GB
Hard Disk	40 GB
Other Hardware	Keyboard, Mouse.

Environment requirements:

Server side
<ul style="list-style-type: none"> • Windows 10 (Any Edition). • Visual Studio 2019 (ASP.NET with c#) • MySql.
Client side

- Operating System : Window 10, XP, Vista, Linux
- Browser : Internet explorer 6.0 onwards/Mozilla Firefox/ Chrome

Chapter III

REQUIREMENT DETERMINATION & ANALYSIS

3.1 Fact finding Methods:

Before starting the actual development of the system, system analyst collects the important information about manual system from the user department. In order to collect this information, system analyst prefers any of the following fact finding techniques.

- Interview
- Questionnaire
- Record review
- Observation

Interview:

Analysts use interview techniques to collect information of system from individuals or smaller groups. The respondent is generally a current user of existing system or potential proposed system. This technique requires some more time than the other fact finding techniques. It is important to remember that the respondent & analyst should conserve only during the interview. Interview techniques allow analyst to discover the area of misunderstanding & unrealized expectation & even indicate of resistance to the proposed system.

Questionnaire:

The use of questionnaire allows the analyst to collect the information about various aspects of system, from large number of person. The use of standardized question format can yield more reliable anonymity for respondent which can lead to more honest responses.

Record review:

Many kinds of records and reports can provide analyst valuable information & operation. In record review analyst examines information that has system & user.

Record exception can be performed at the beginning of the study as an introduction or later in the study as a basis for comparing actual operation with for the record indicates should be happening.

Observation:

Observation allows analysts to collect the information, they cannot obtain by other fact finding techniques. Thorough observation analyst can obtain first information about how activities are carried out. This method is most useful when analyst need to actually observe how documents are handled, how processes are carried out & weather the specific steps is actually followed.

3.2 Feasibility Study:

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

Feasibility studies also can provide a company's management with crucial information that could prevent the company from entering carelessly into risky businesses.

A feasibility study in project management usually assesses the following areas:

- 1.**Technical capability:** Does the organization have the technical resources to undertake the project?
- 2.**Budget:** Does the organization have the financial resources to undertake the project, and is the cost/benefit analysis sufficient to warrant moving forward?
- 3.**Legality:** What are the legal requirements of the project, and can the business meet them?

4.Risk: What is the risk associated with undertaking this project? Is the risk worthwhile to the company based on perceived benefits?

5.Operational feasibility: Does the project, in its proposed scope, meet the organization's needs by solving problems and/or taking advantage of identified opportunities?

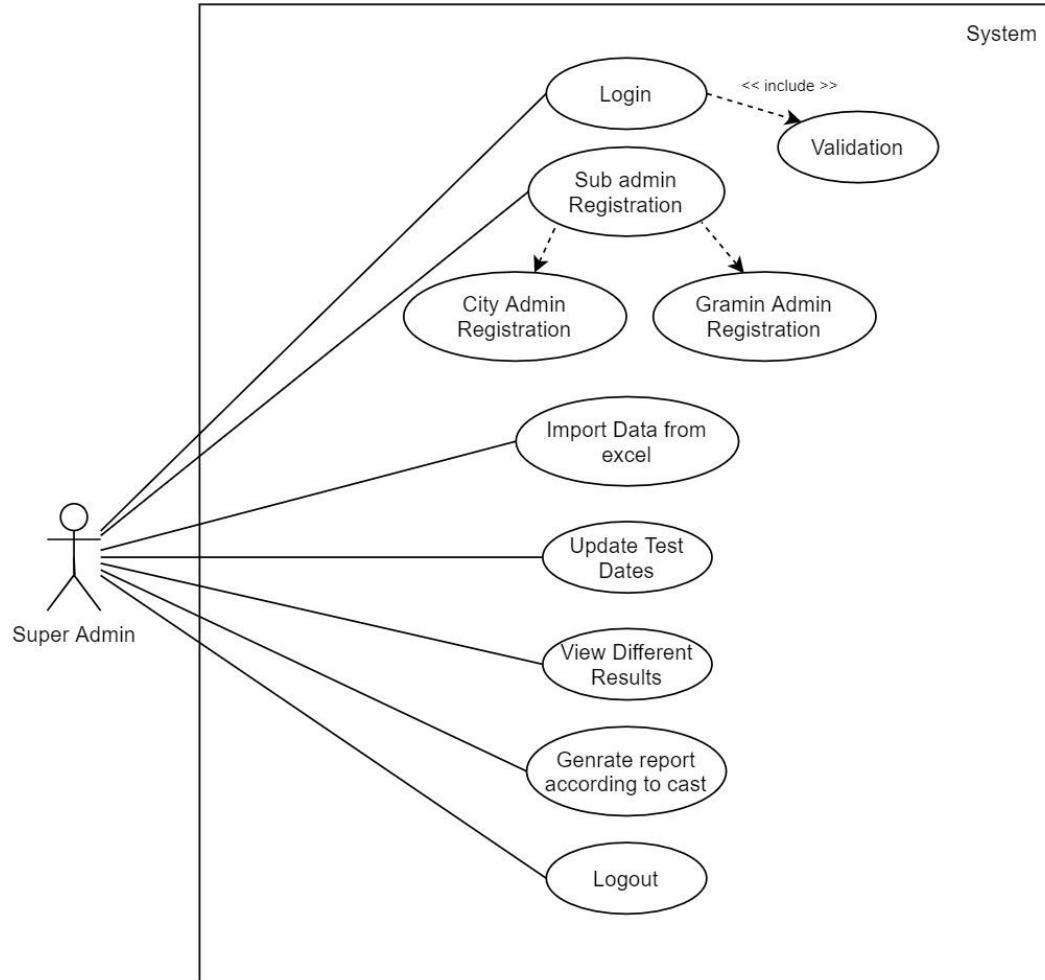
6.Time: Can the project be completed in a reasonable timeline?

Chapter IV

SYSTEM ANALYSIS & DESIGN

4.1 Use case Diagram:

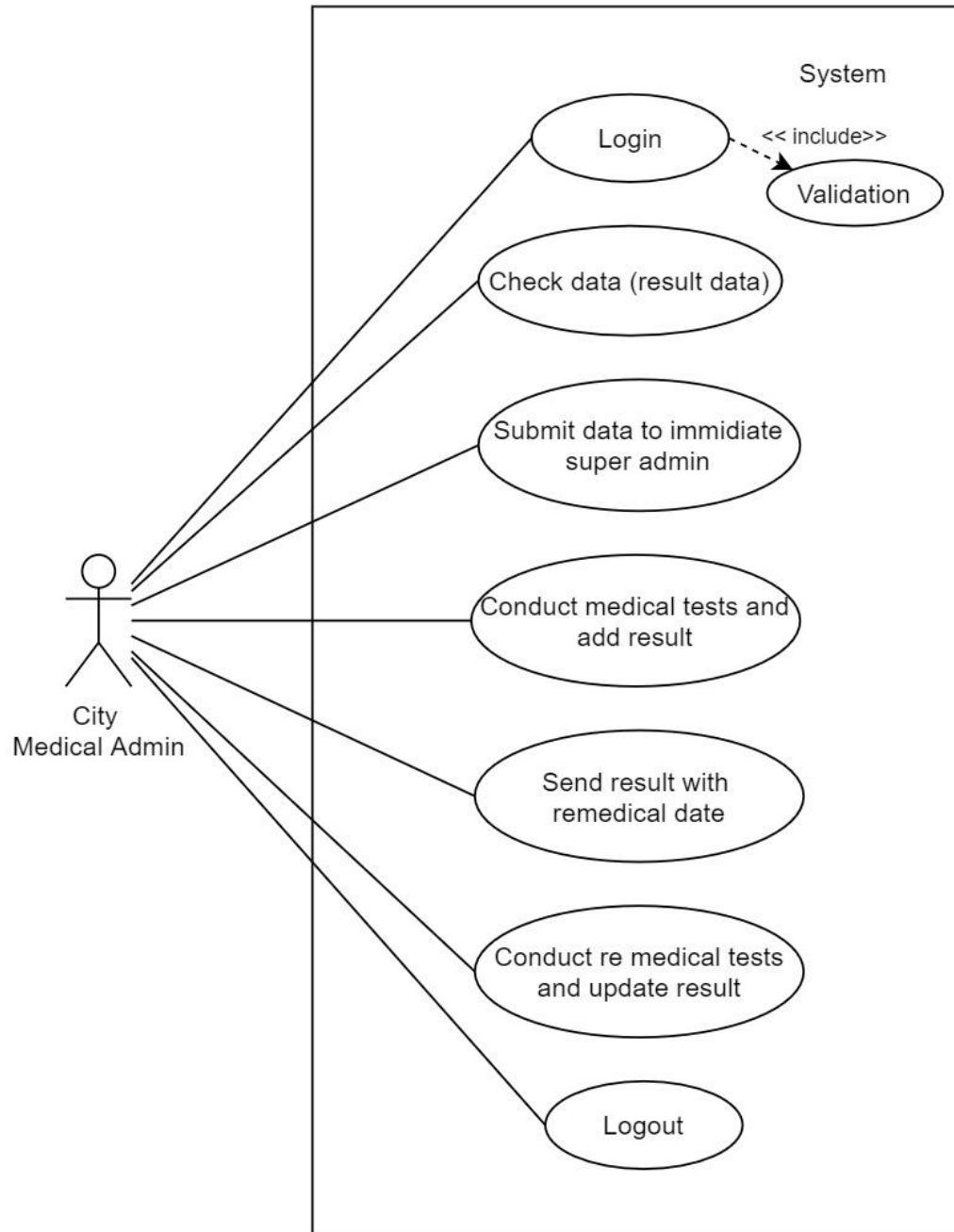
District Admin:



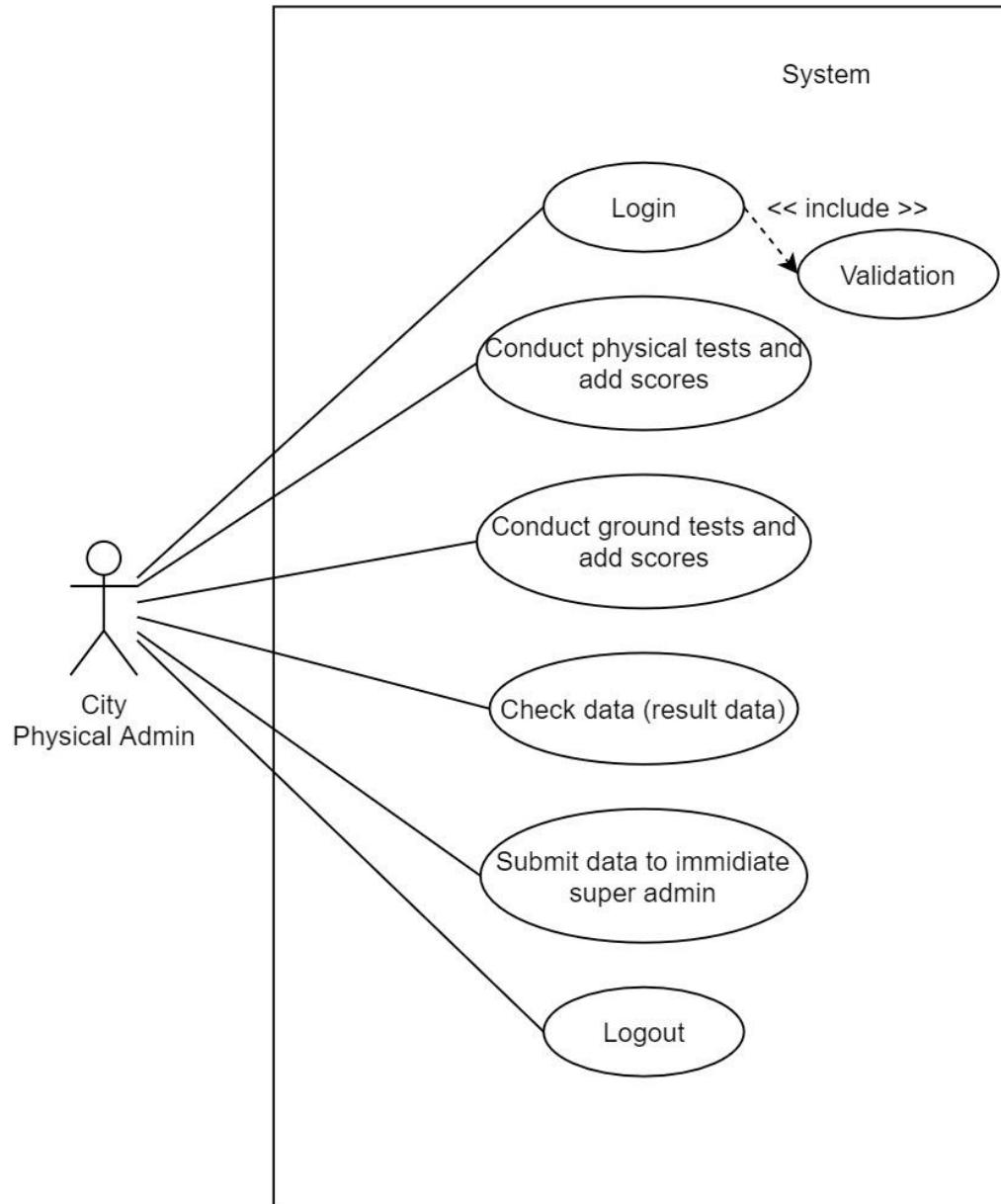
City Admin:



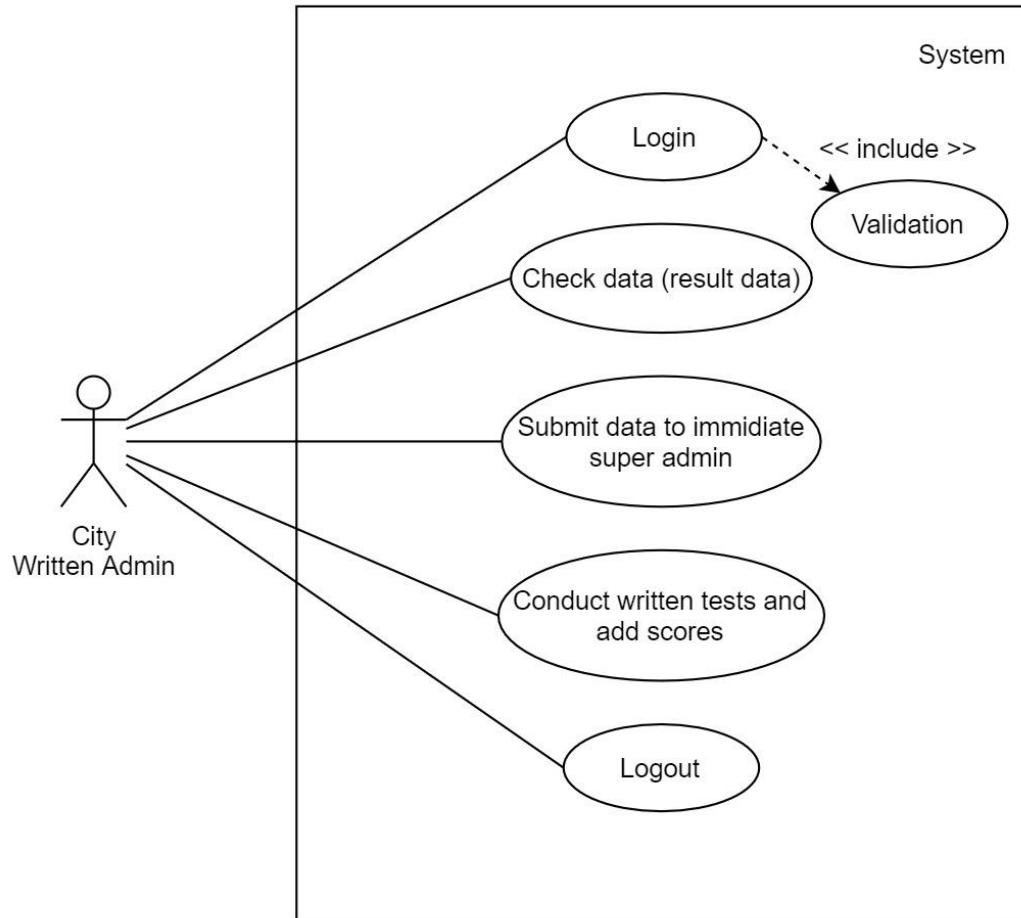
City Medical:



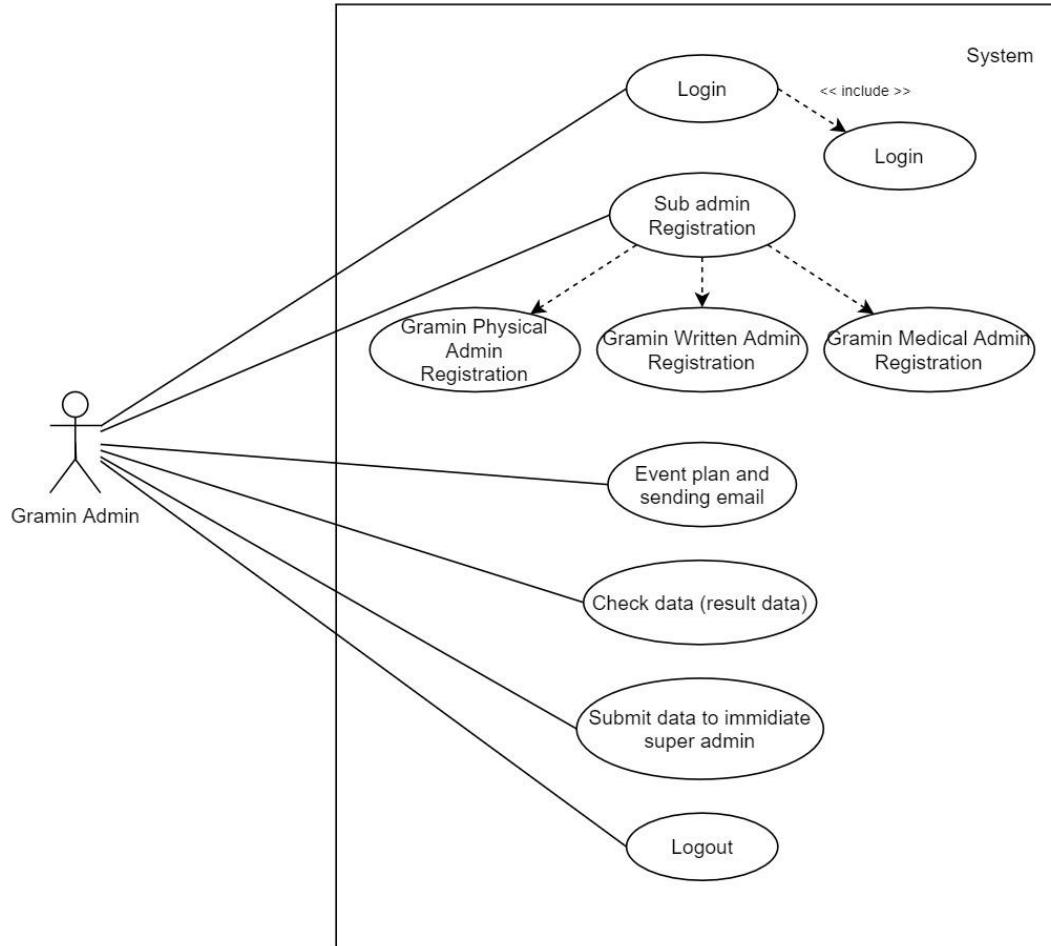
City Physical:



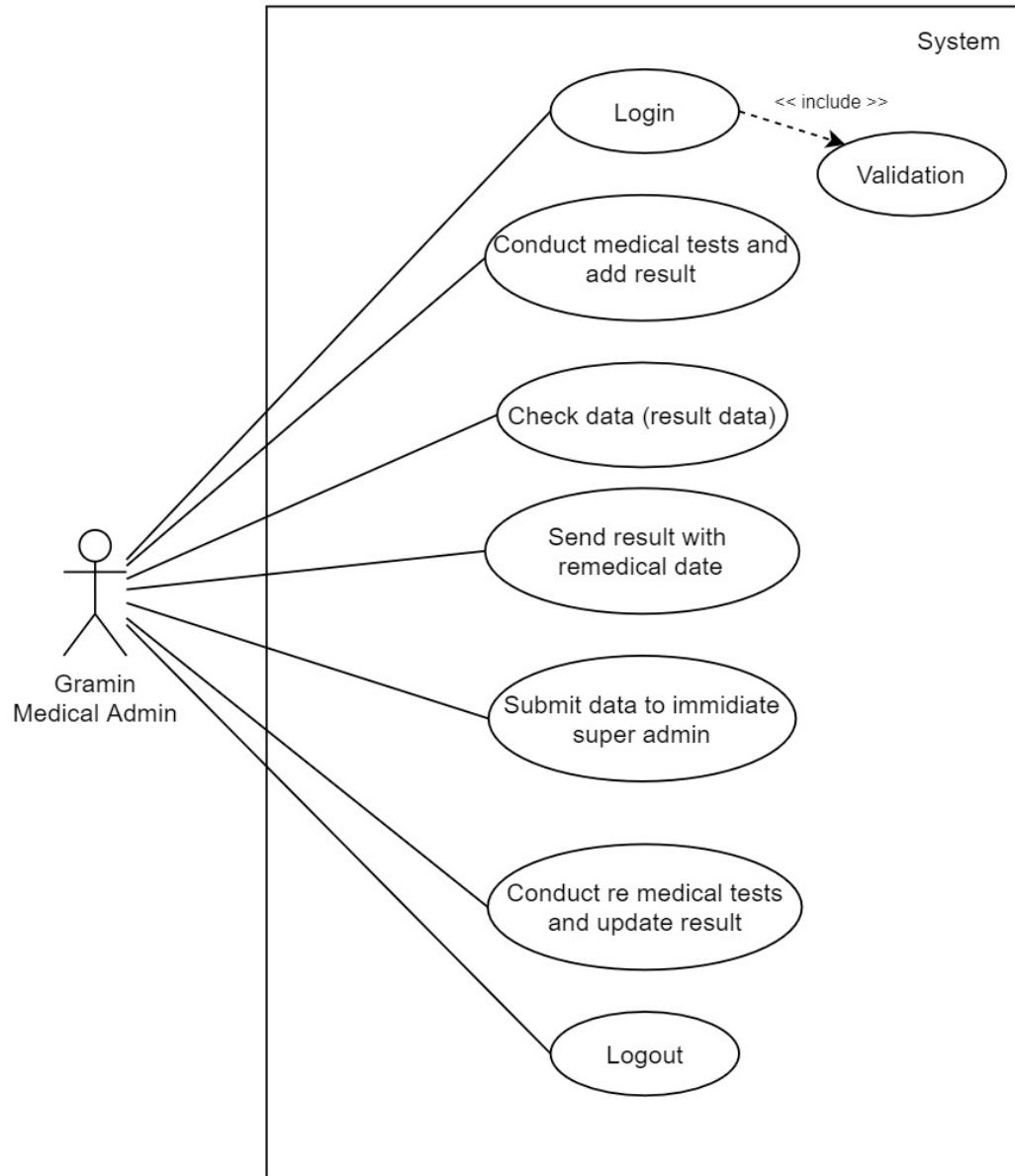
City Written:



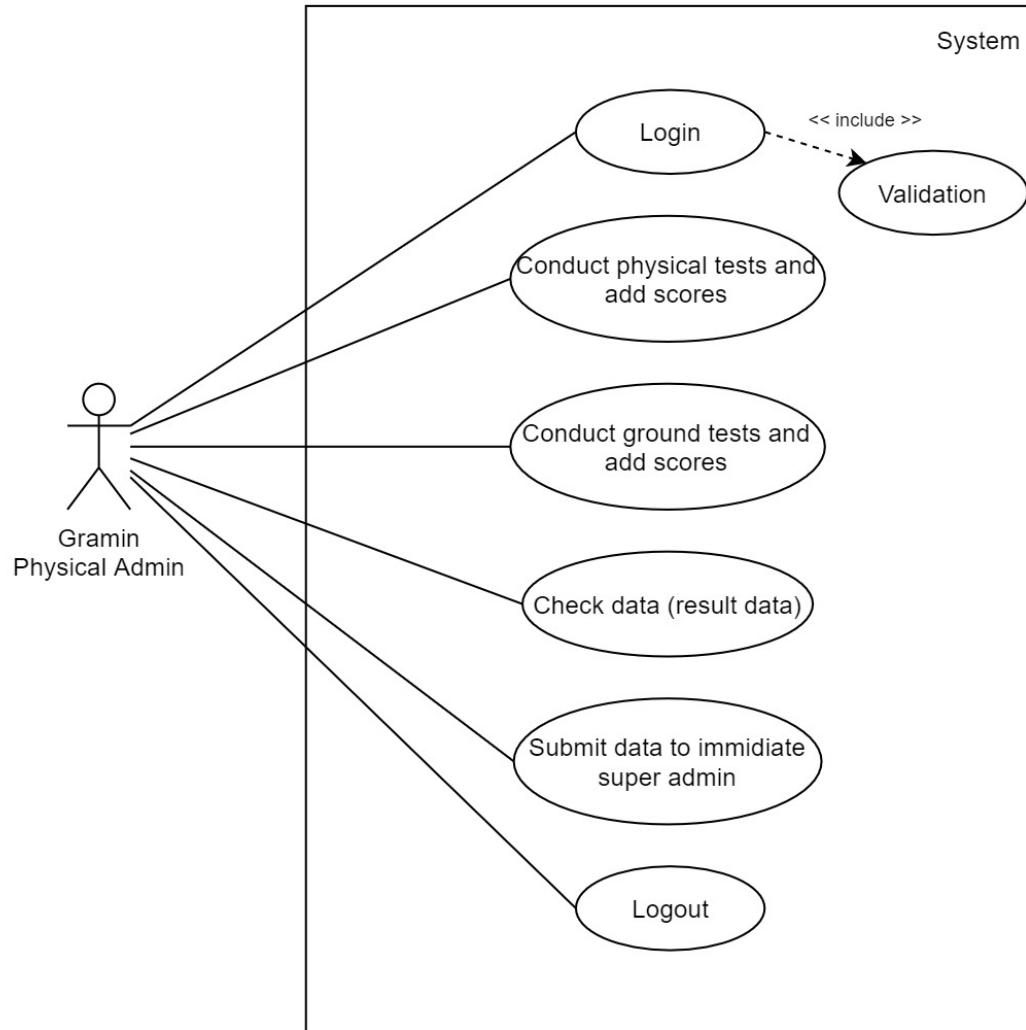
Gramin Admin:



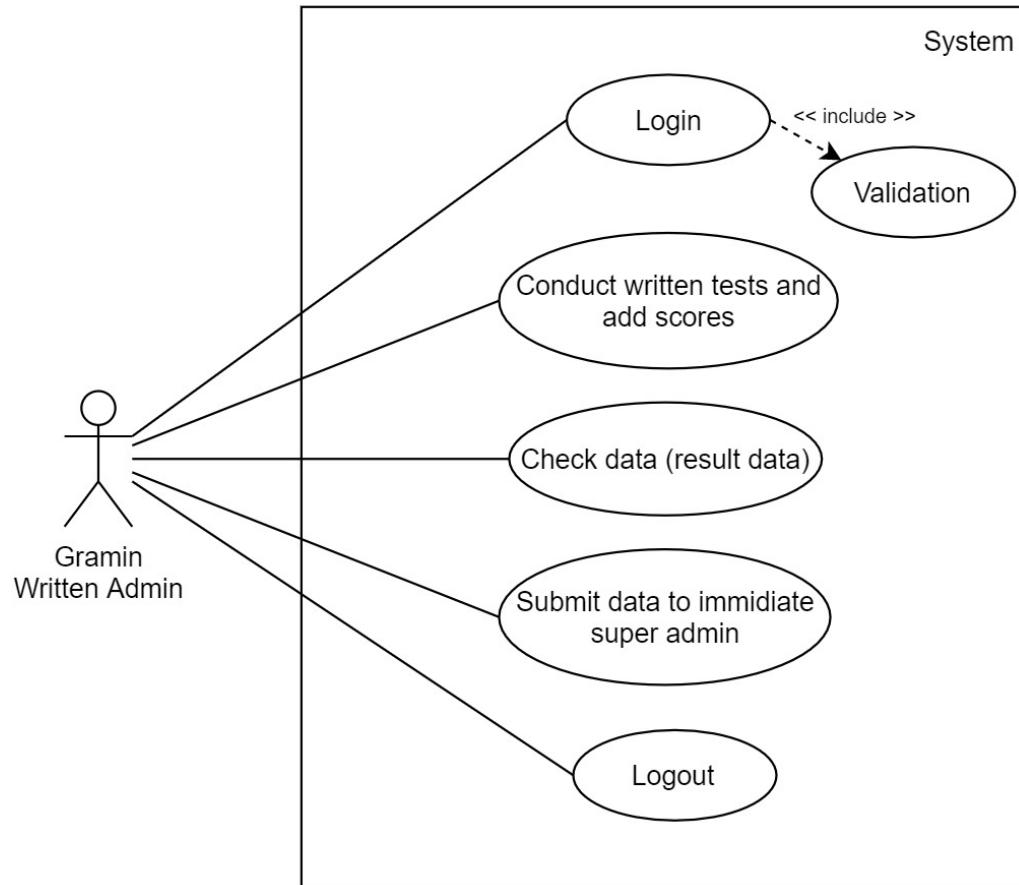
Gramin Medical:



Gramin Physical:

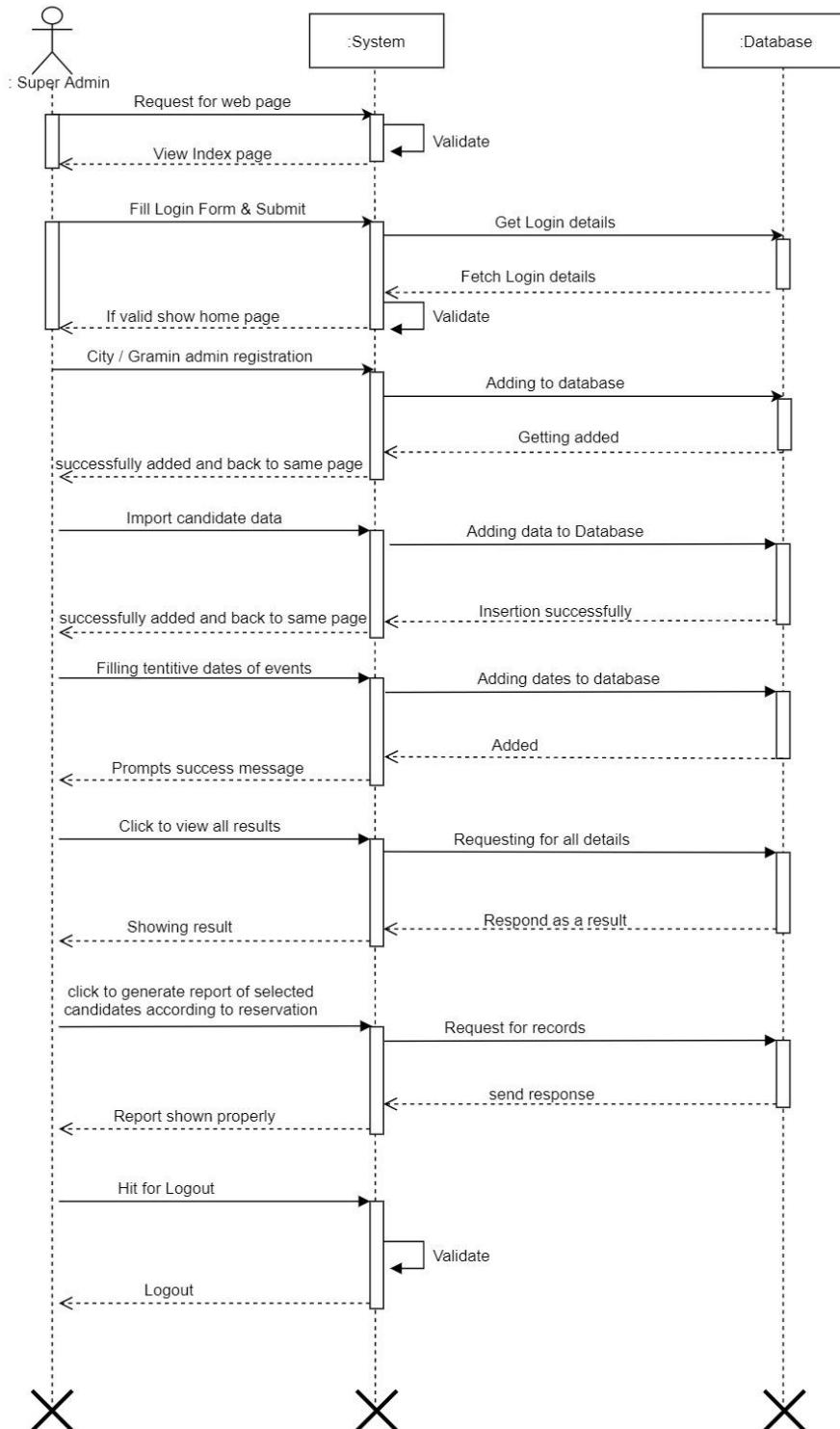


Gramin Written:

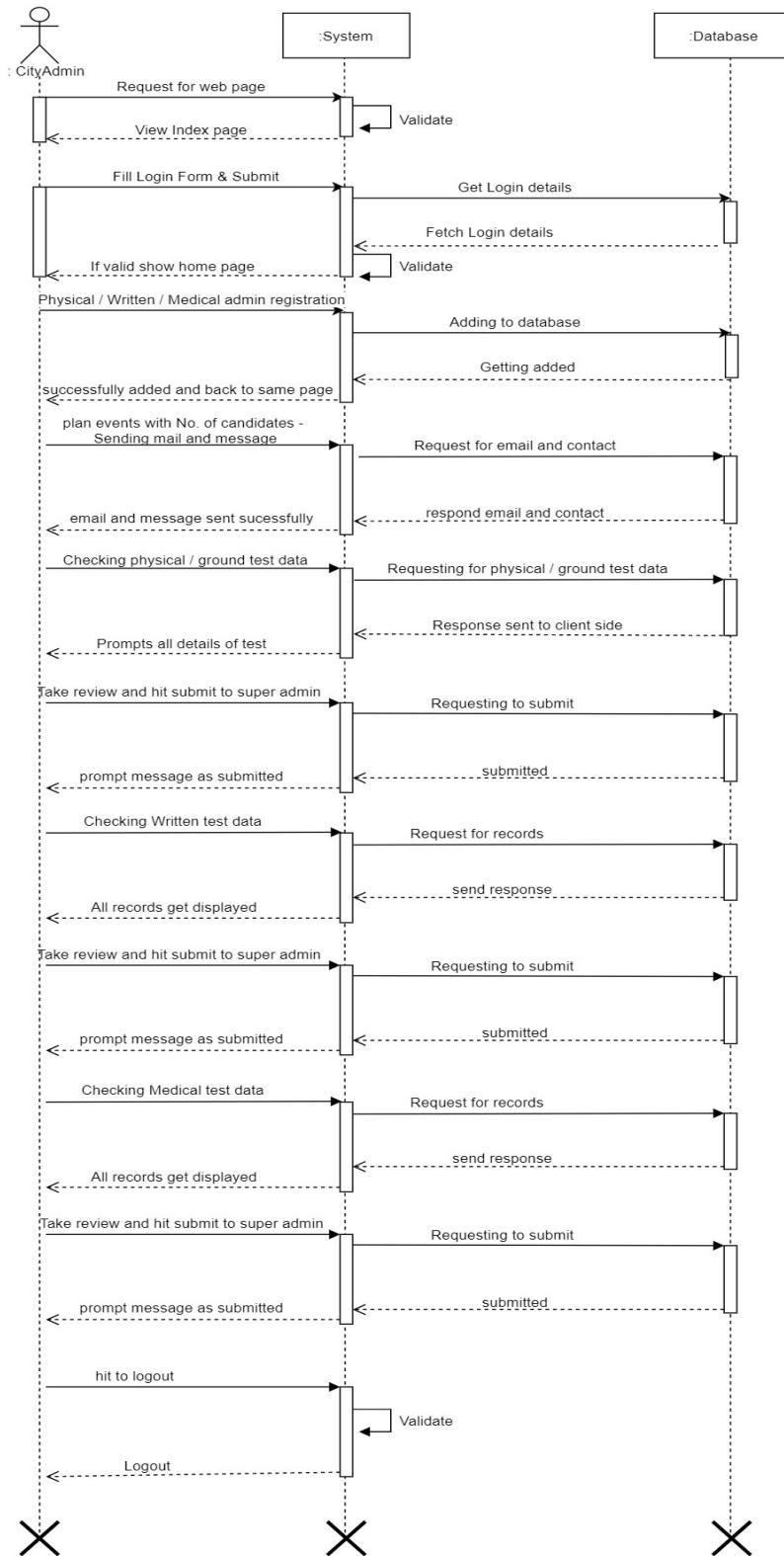


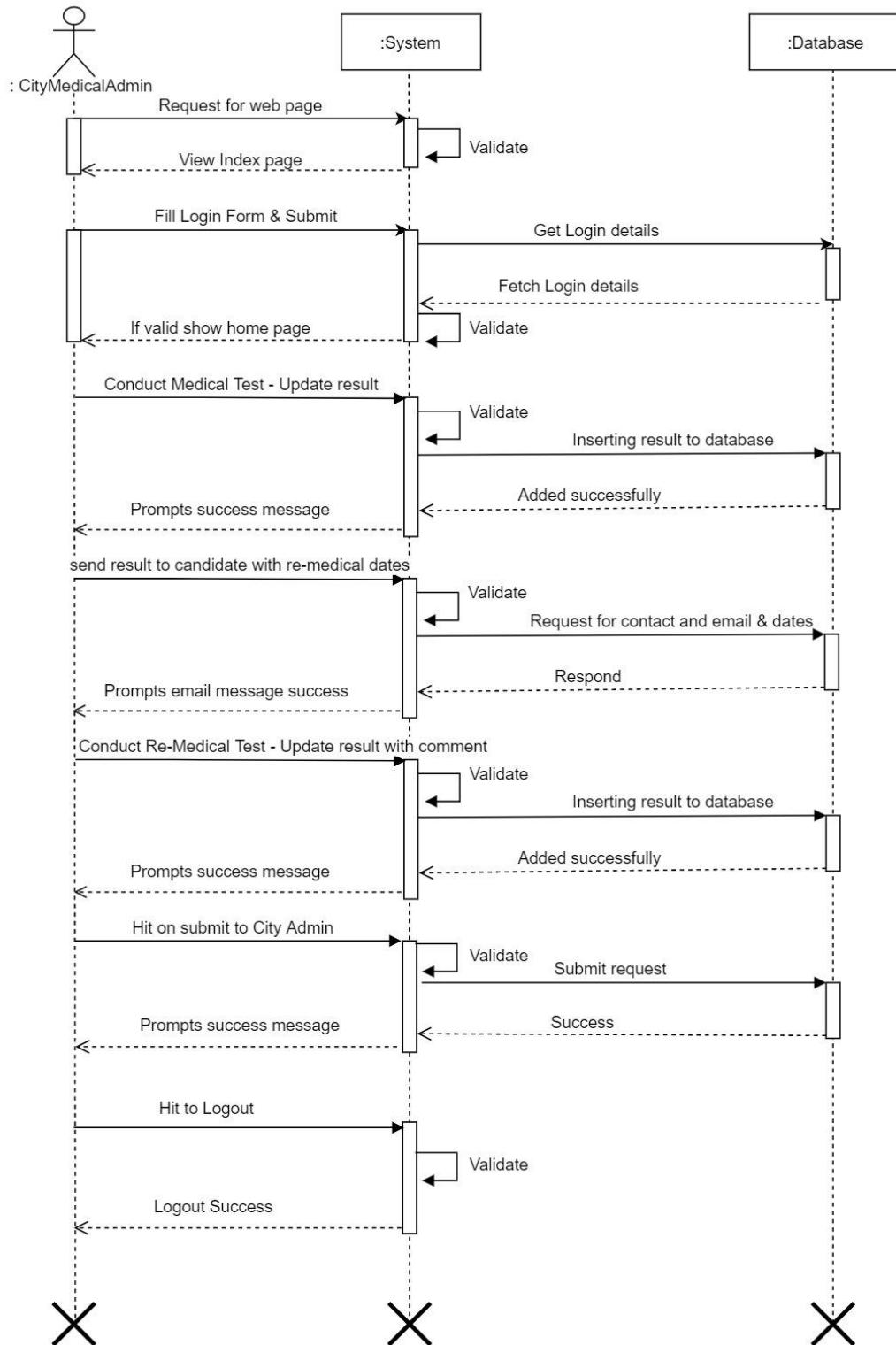
4.2 Sequence Diagram:

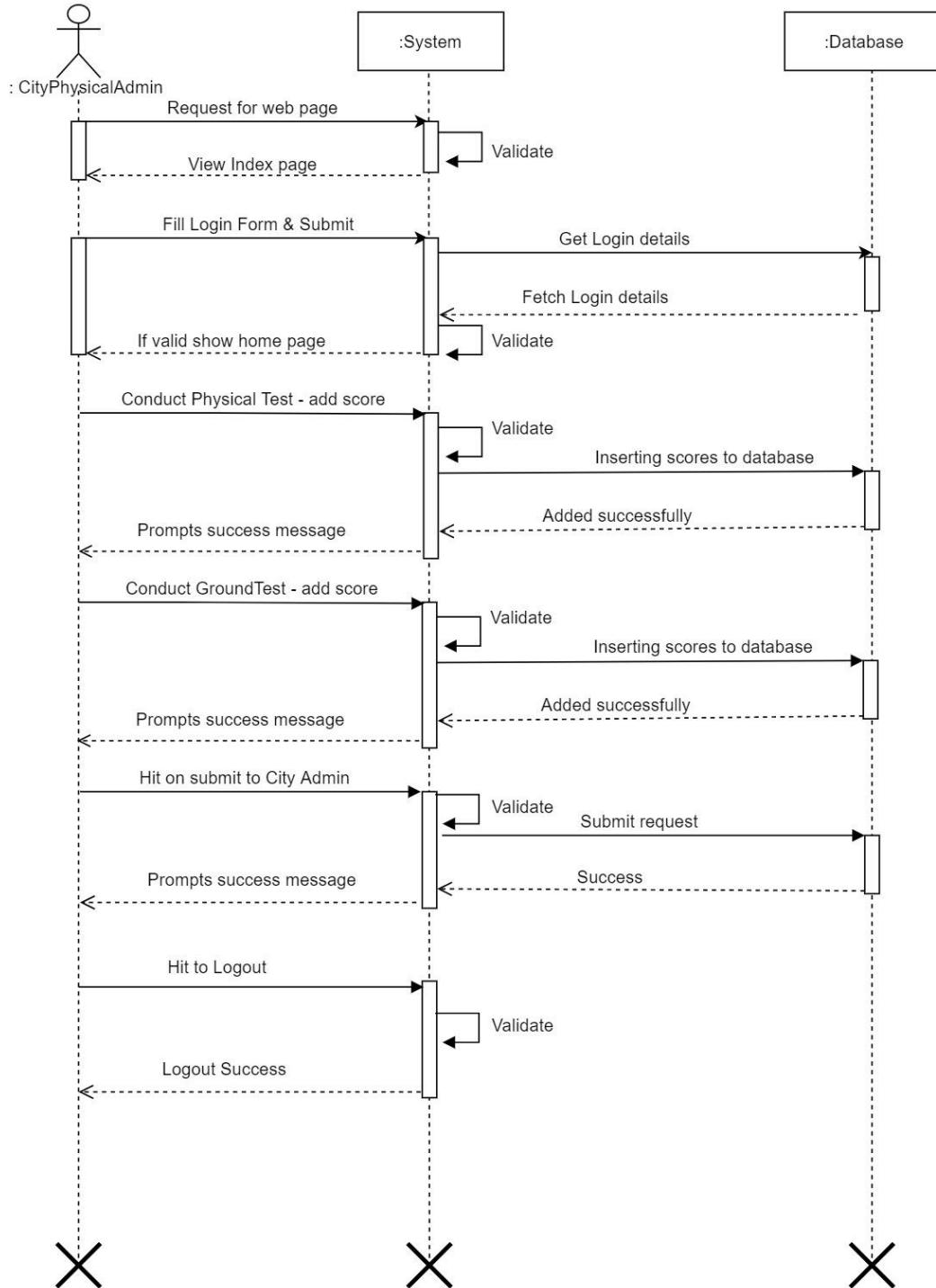
District Admin

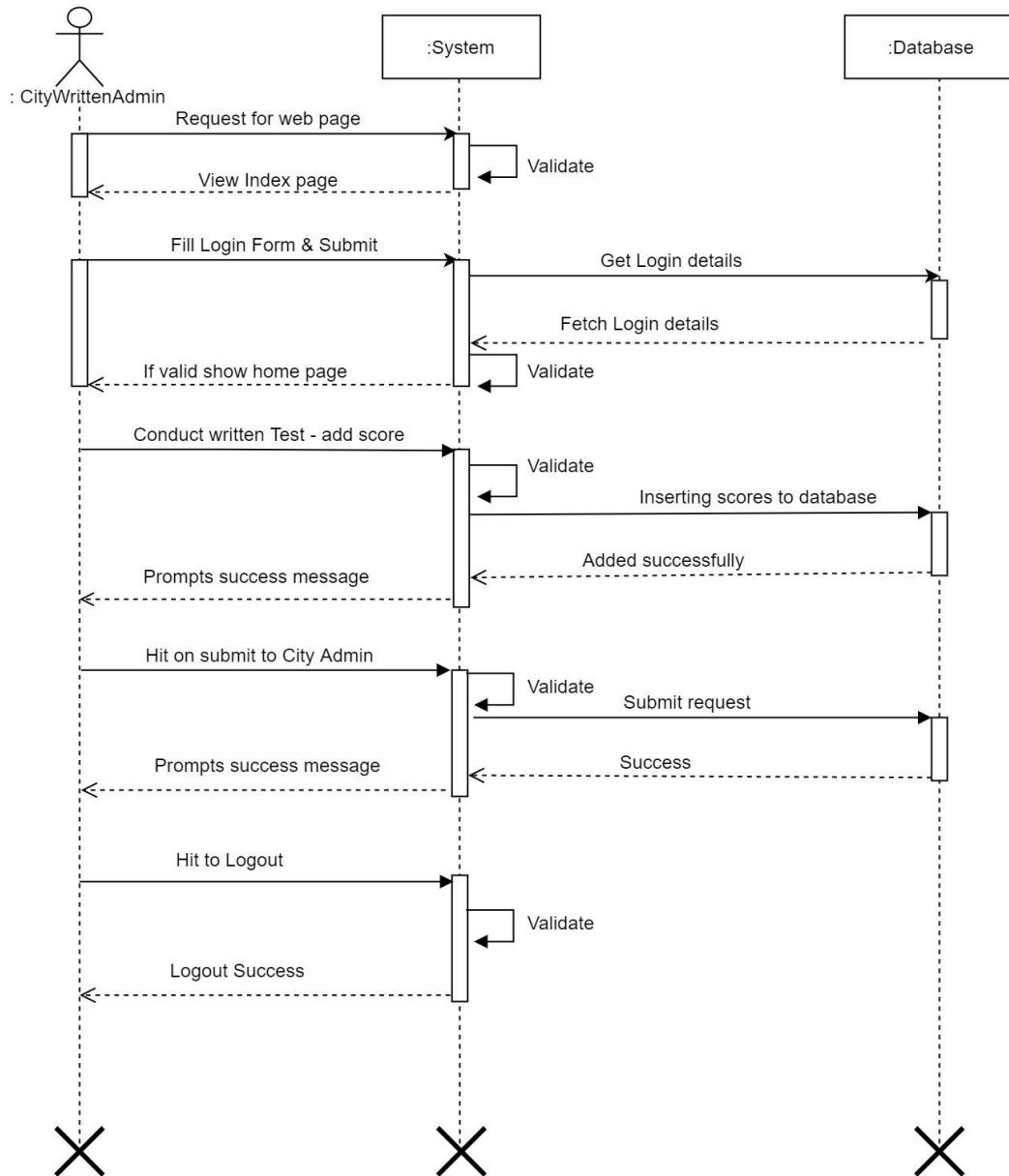


City Admin:

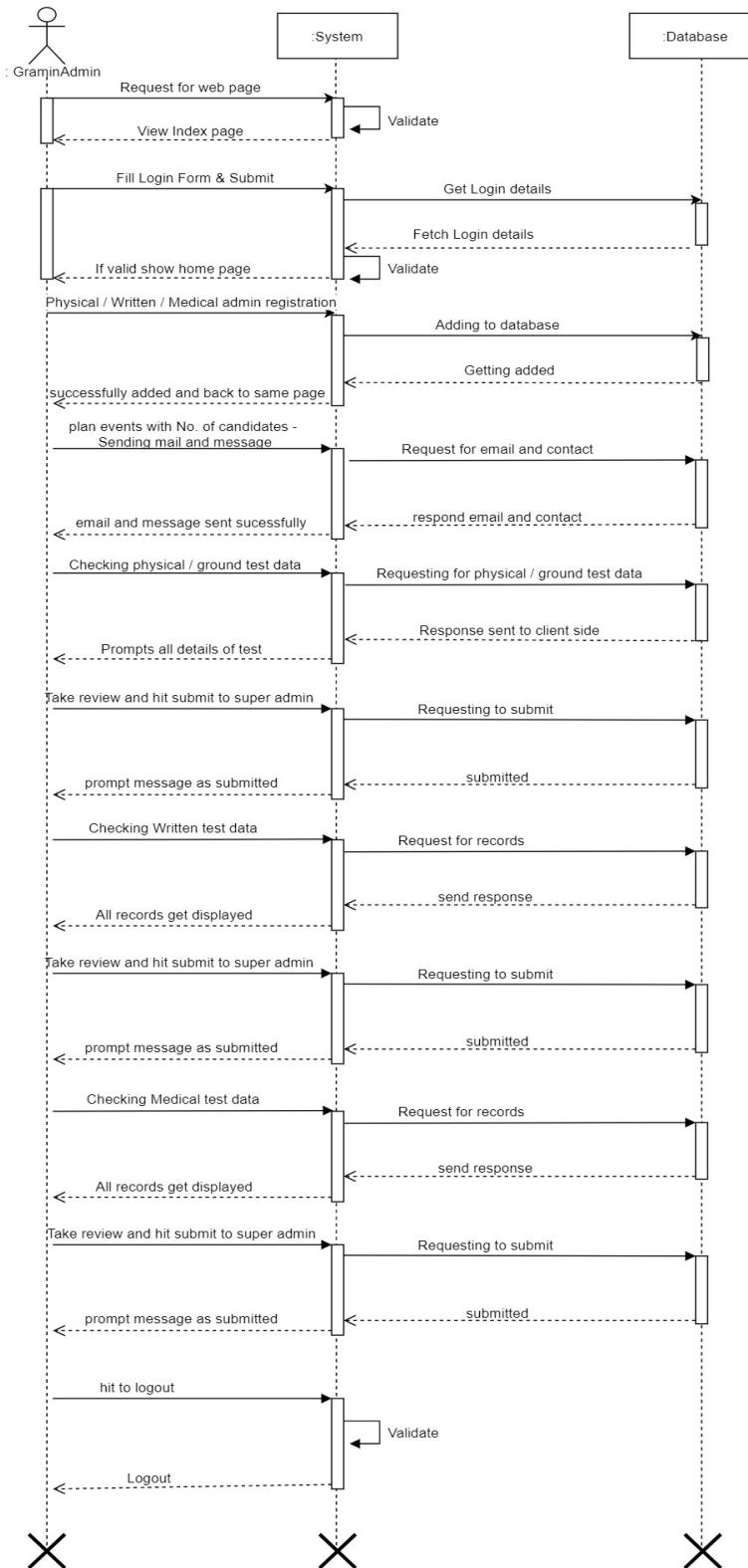


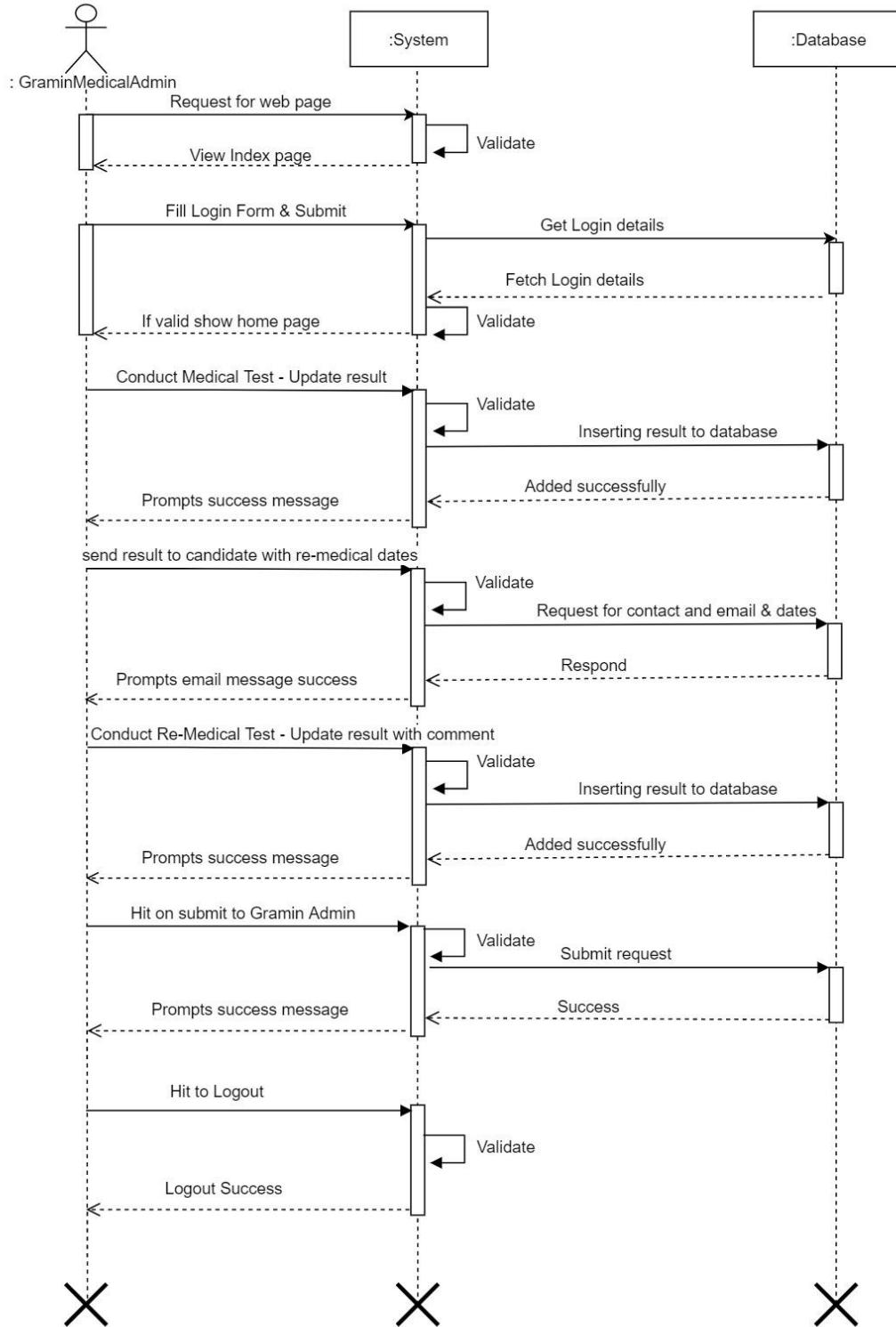
City Medical:

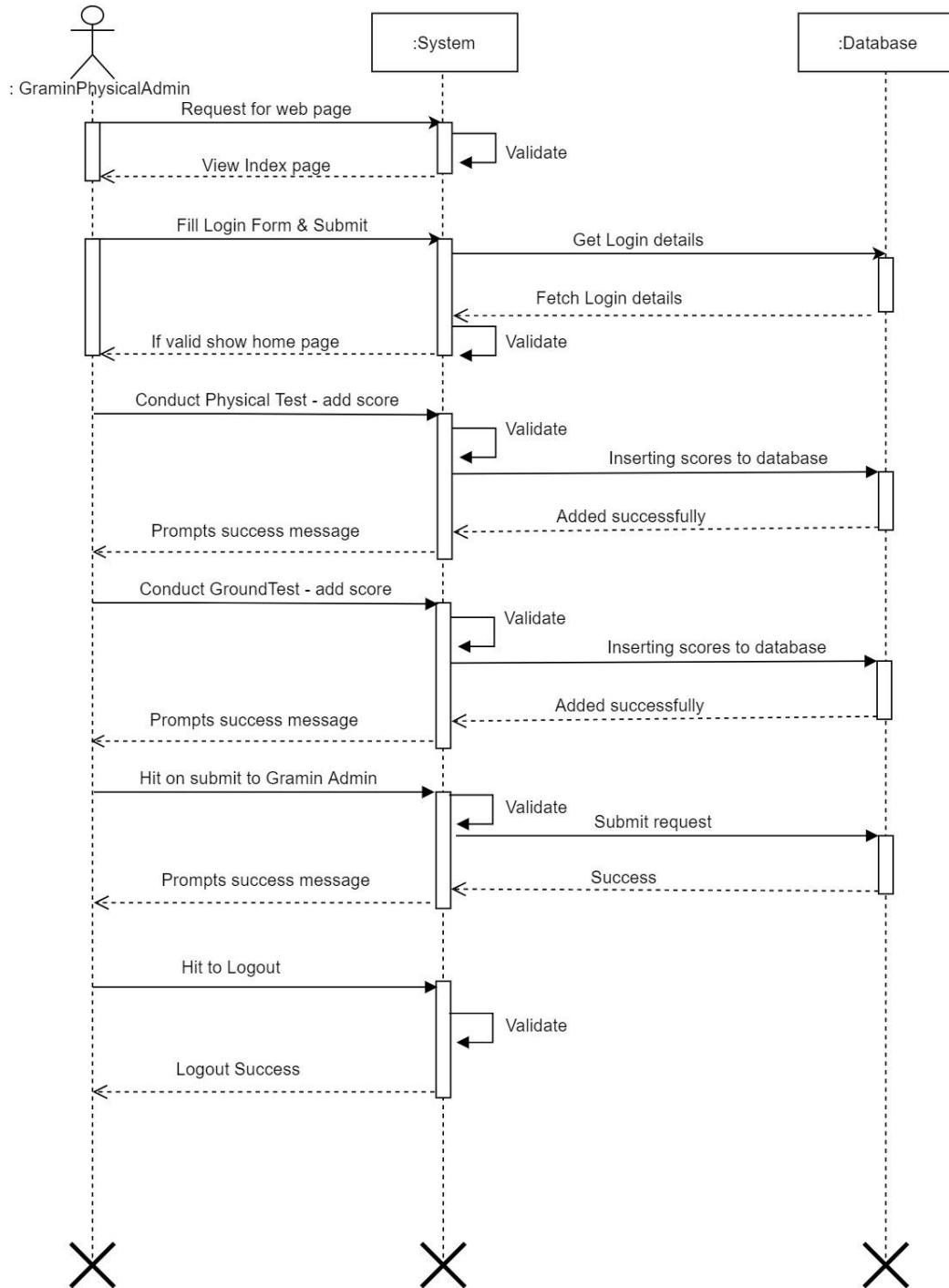
City Physical:

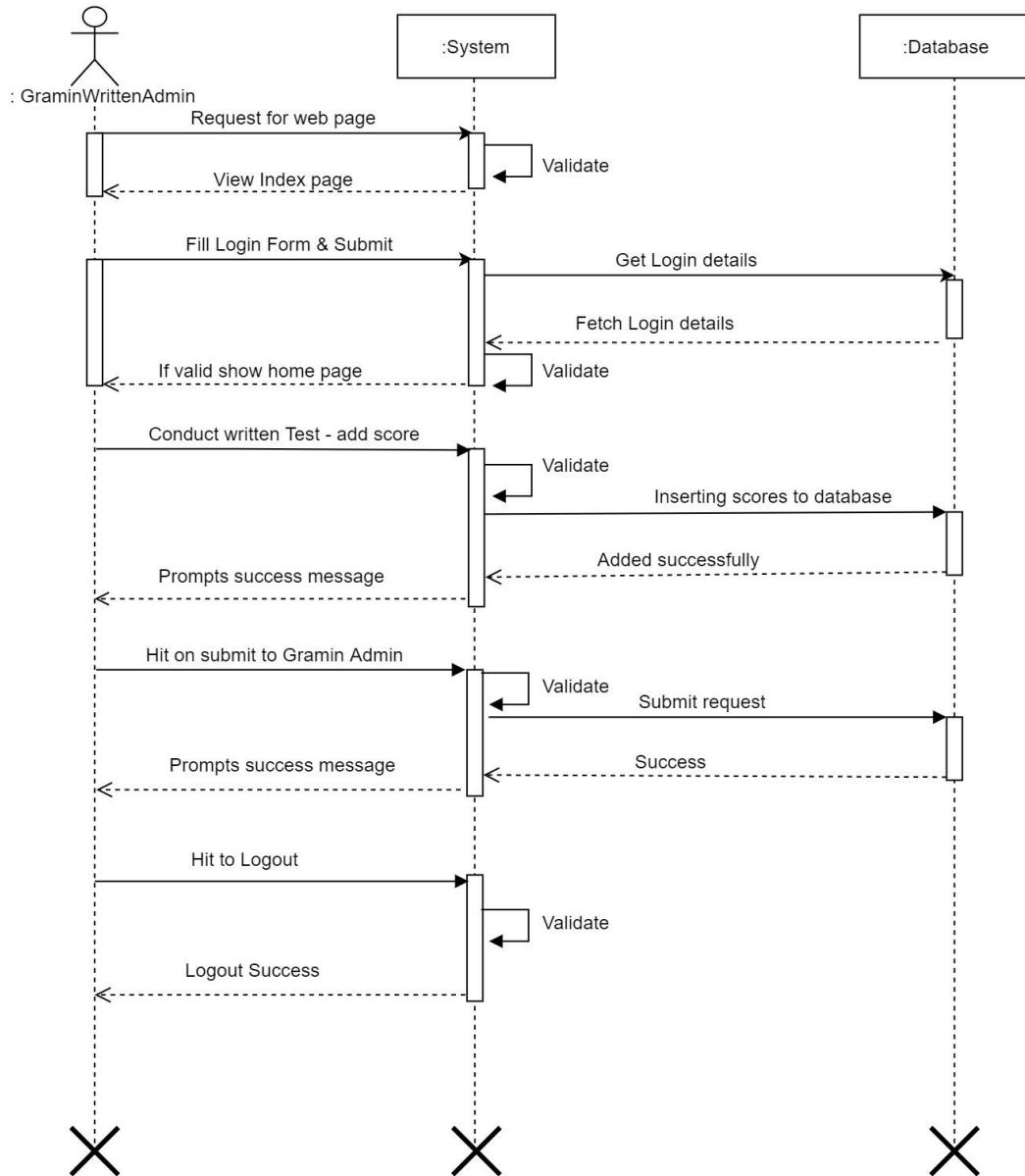
City Written:

Gramin Admin:



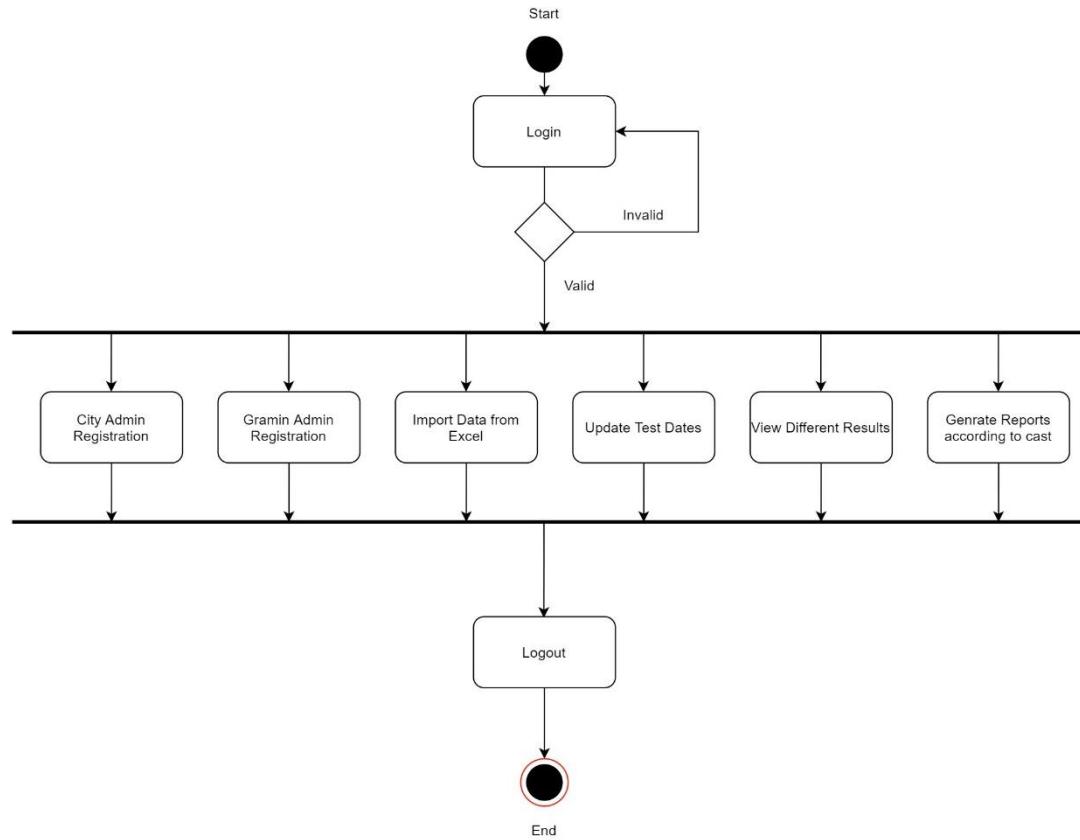
Gramin Medical:

Gramin Physical:

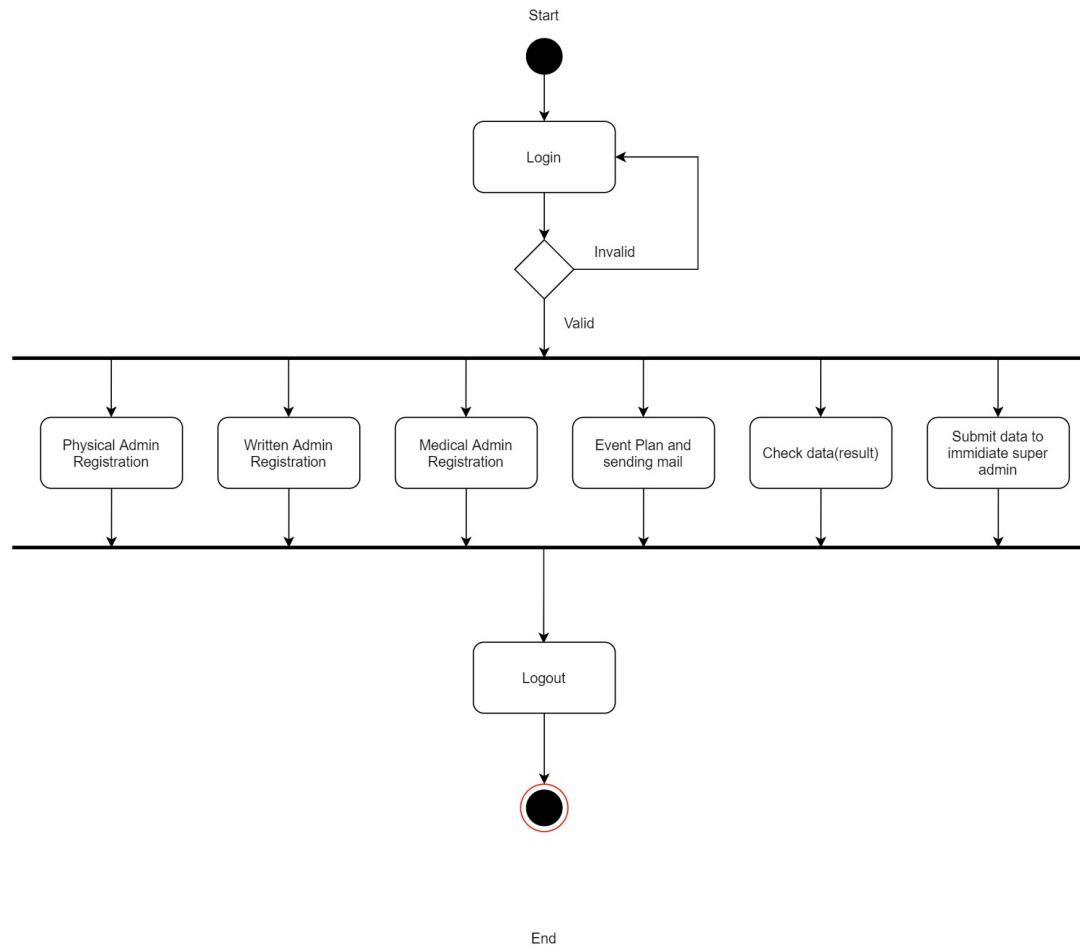
Gramin Written:

4.3 Activity:

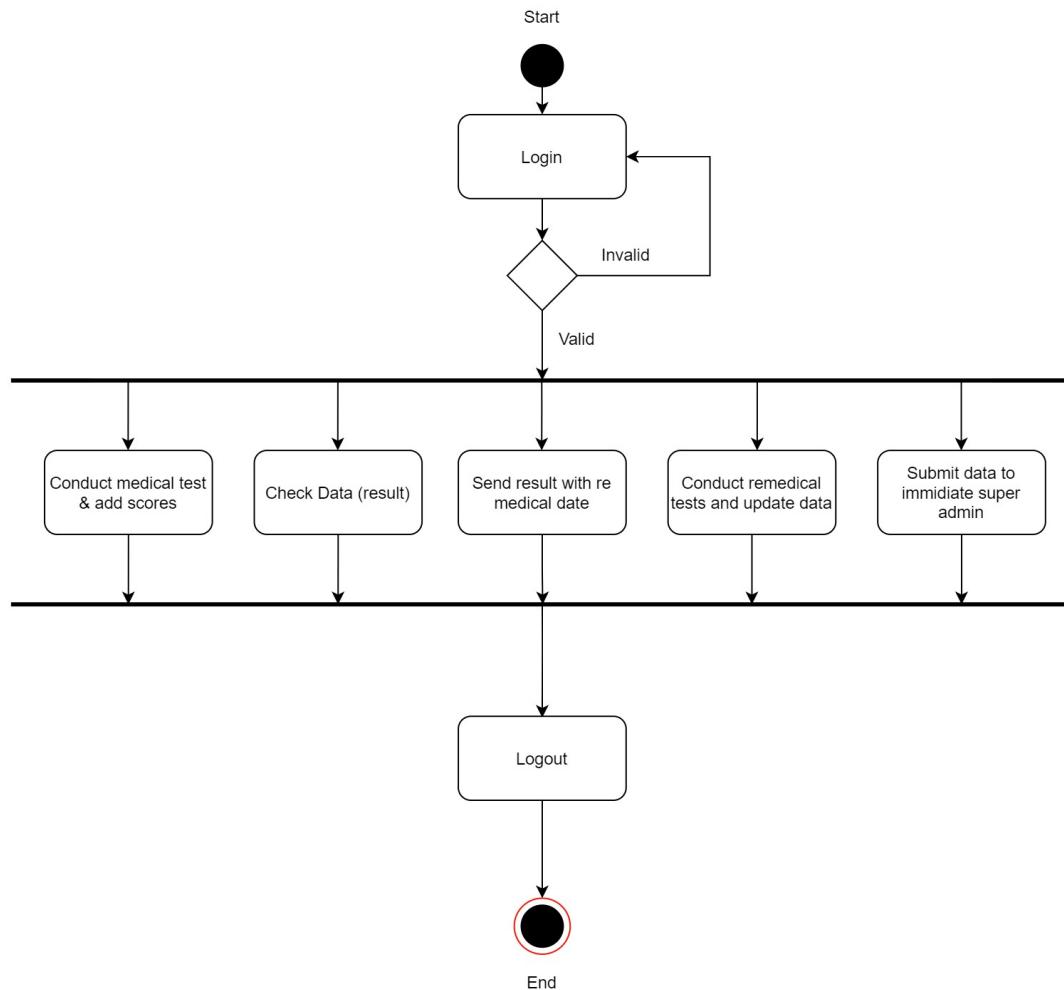
District Admin:



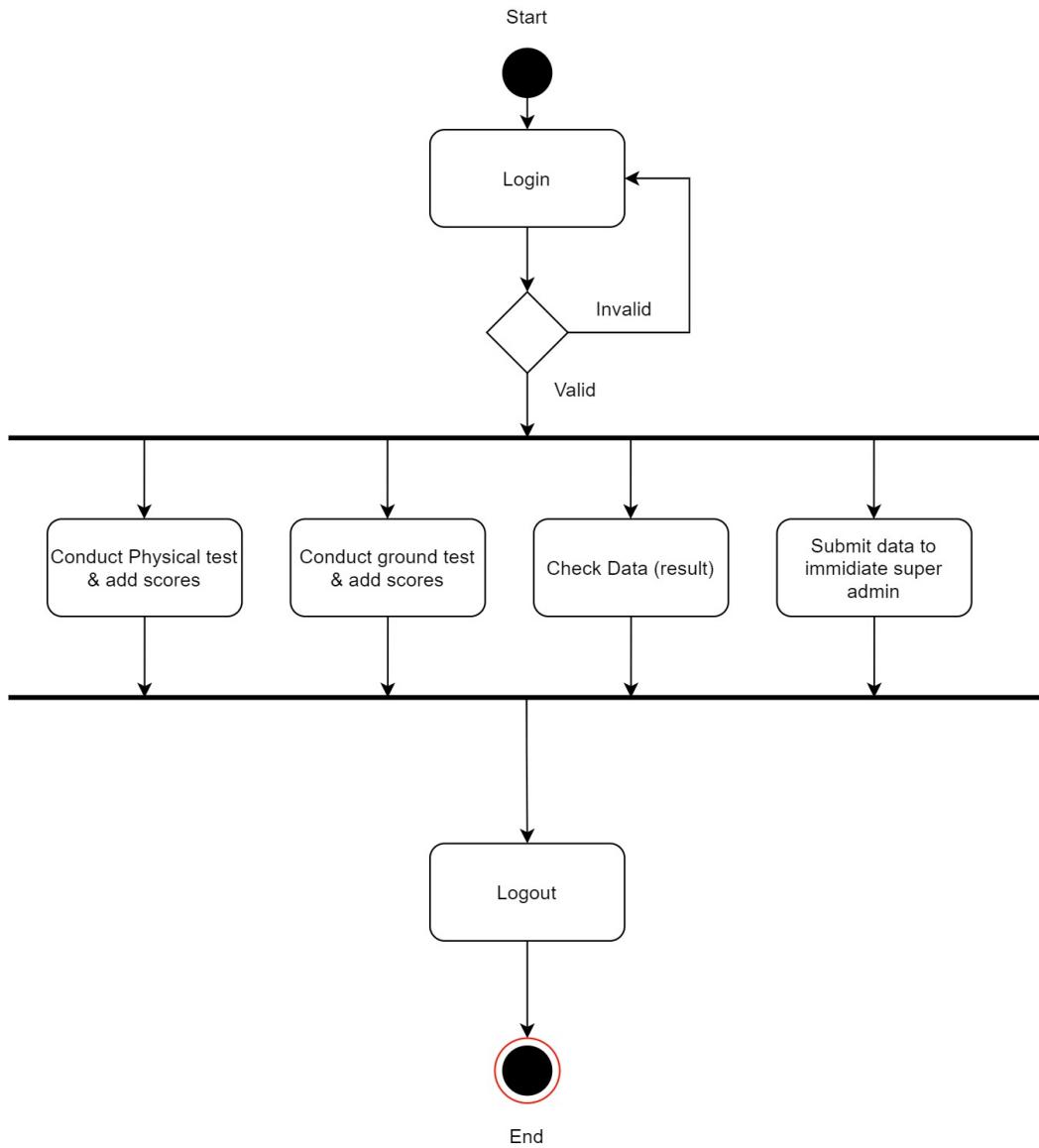
City Admin:



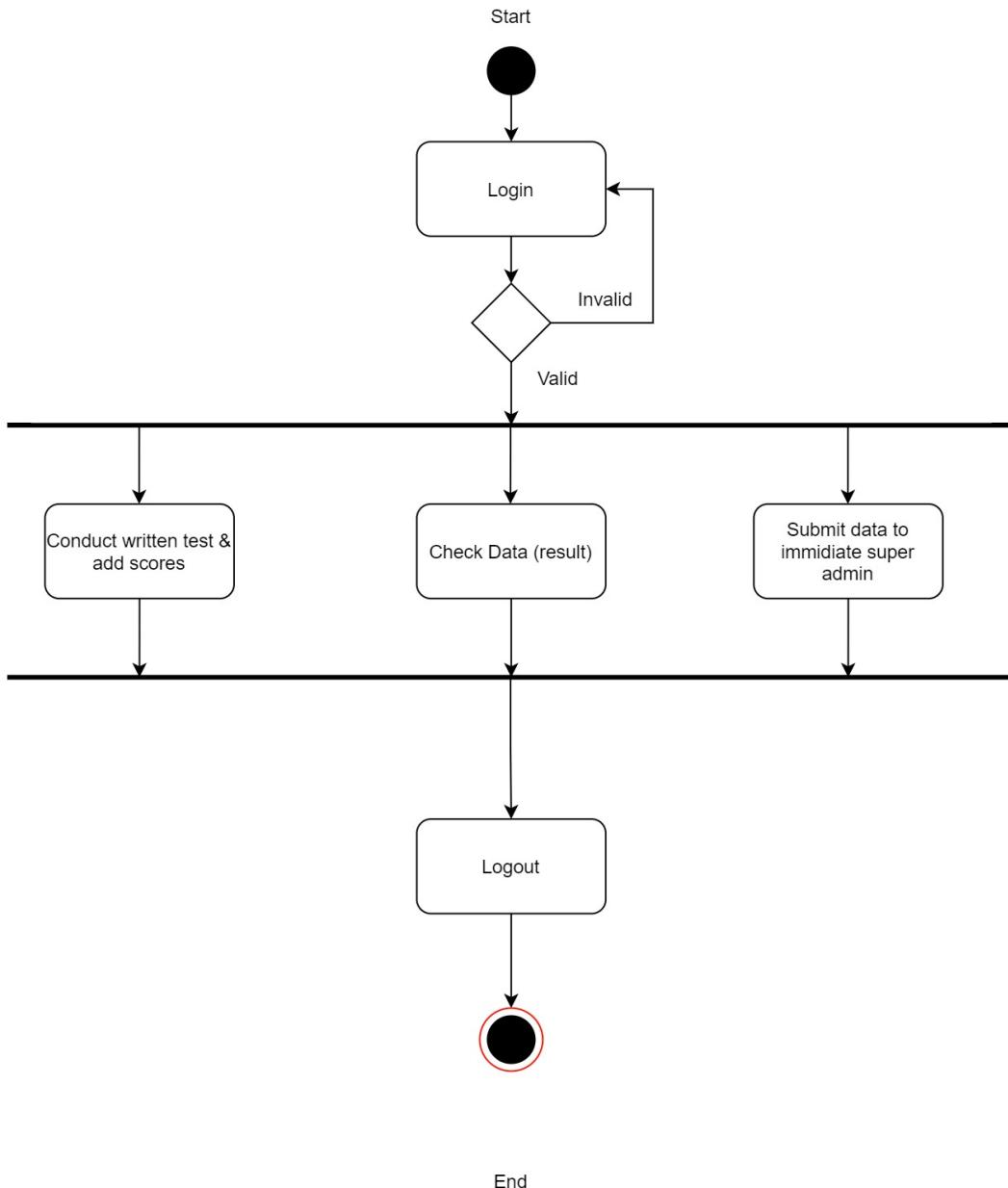
City Medical:



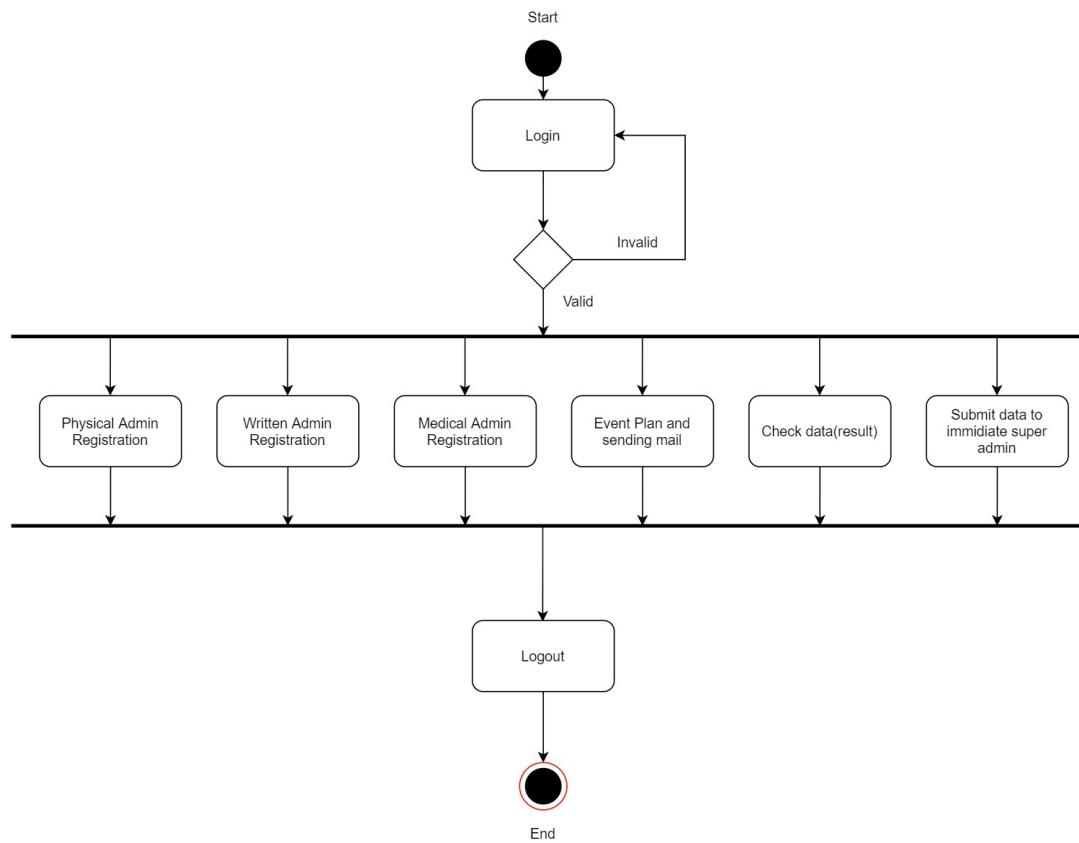
City Physical:



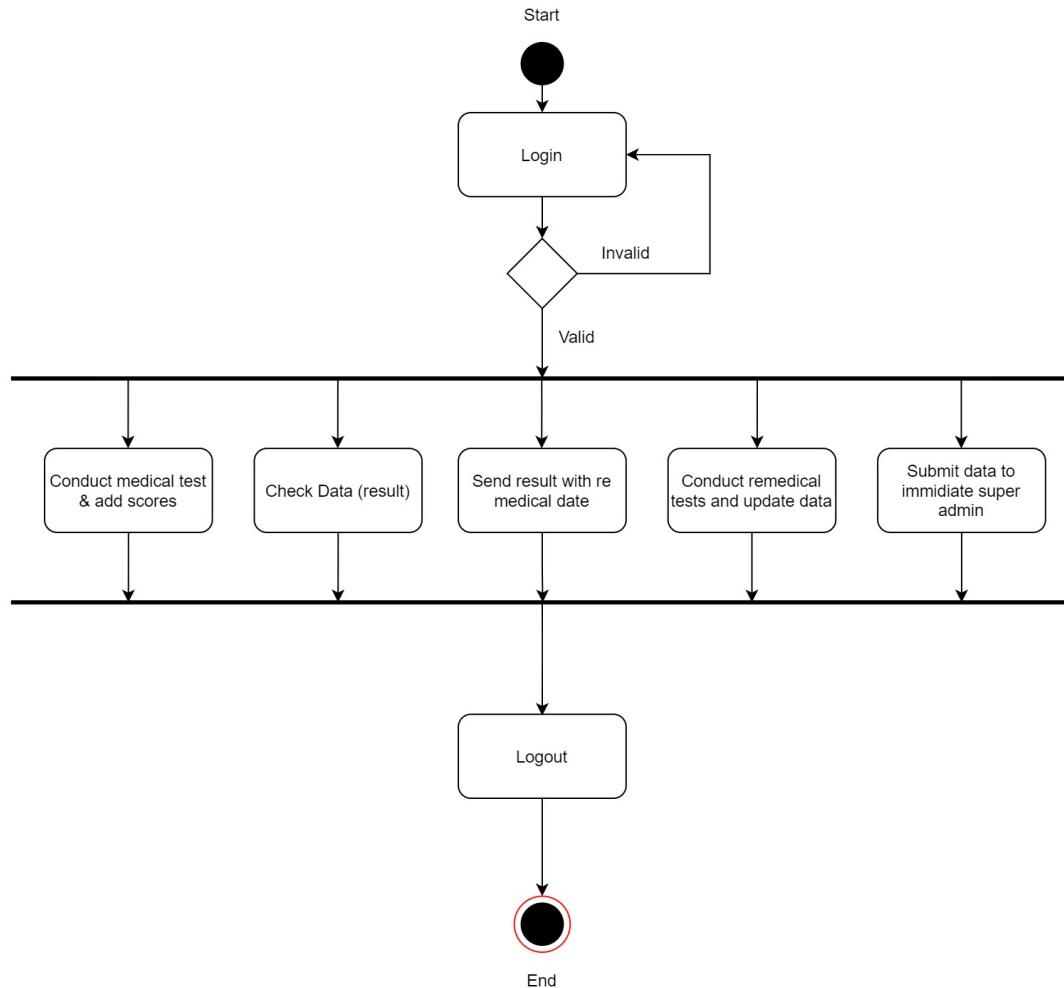
City Written:



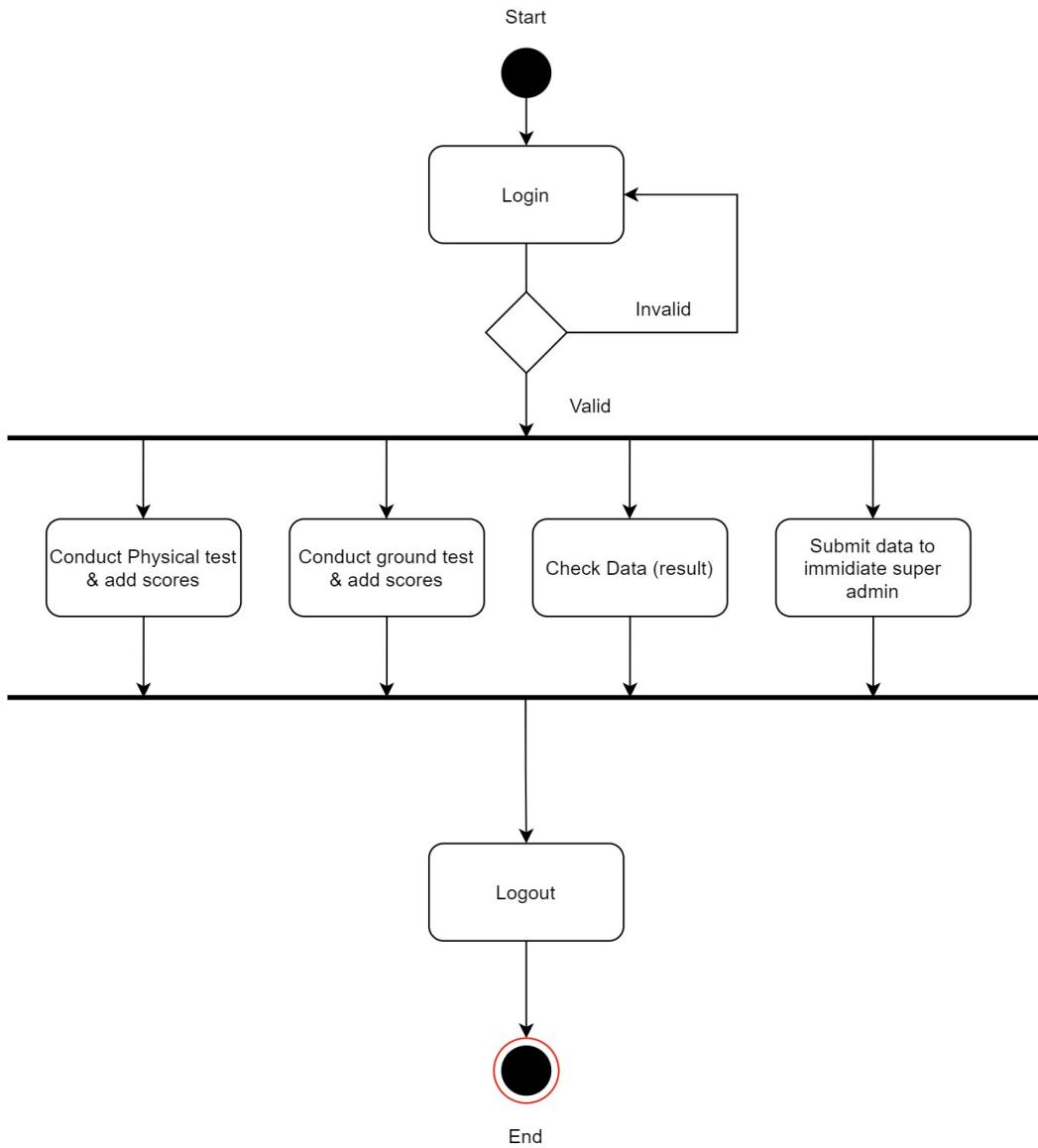
Gramin Admin:



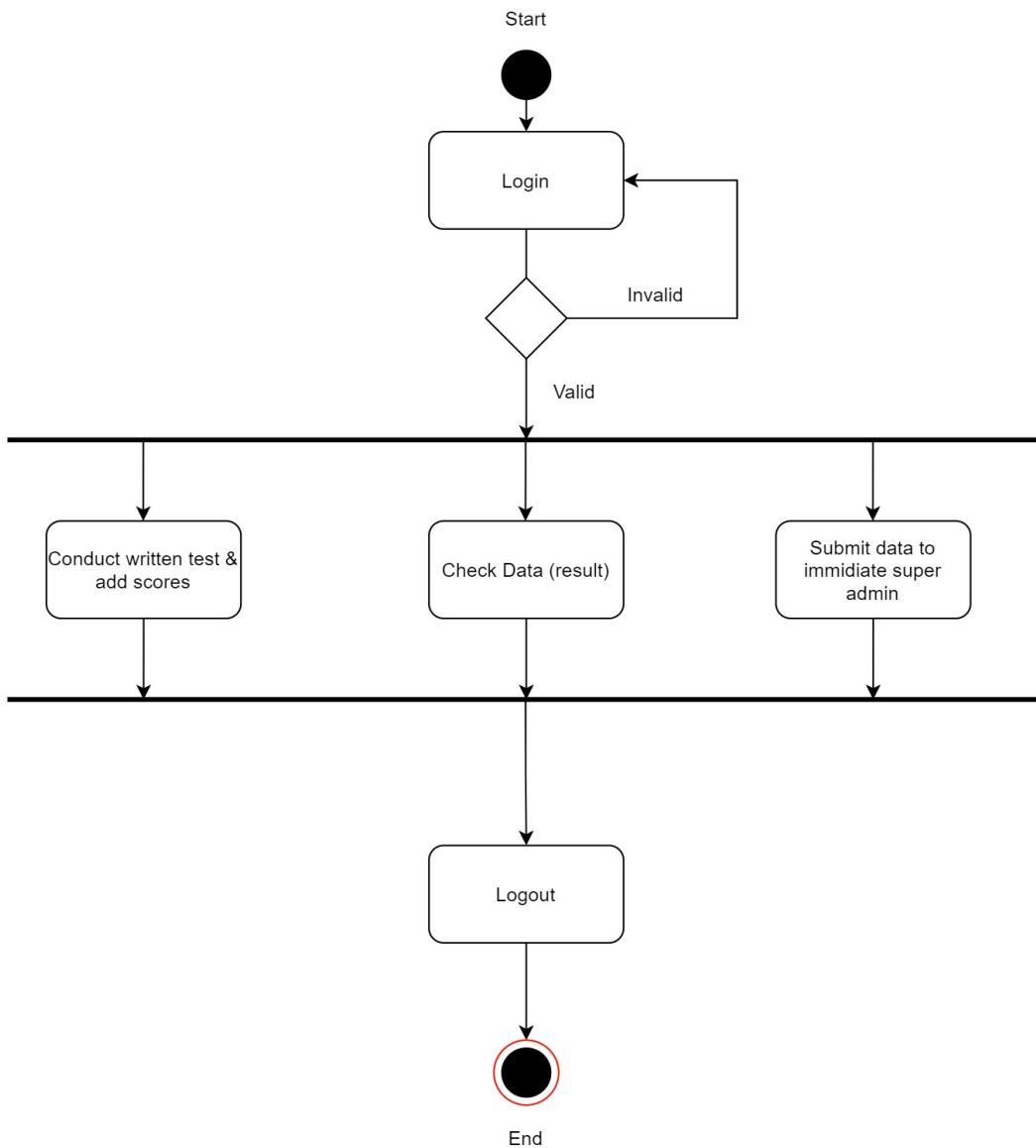
Gramin Medical:



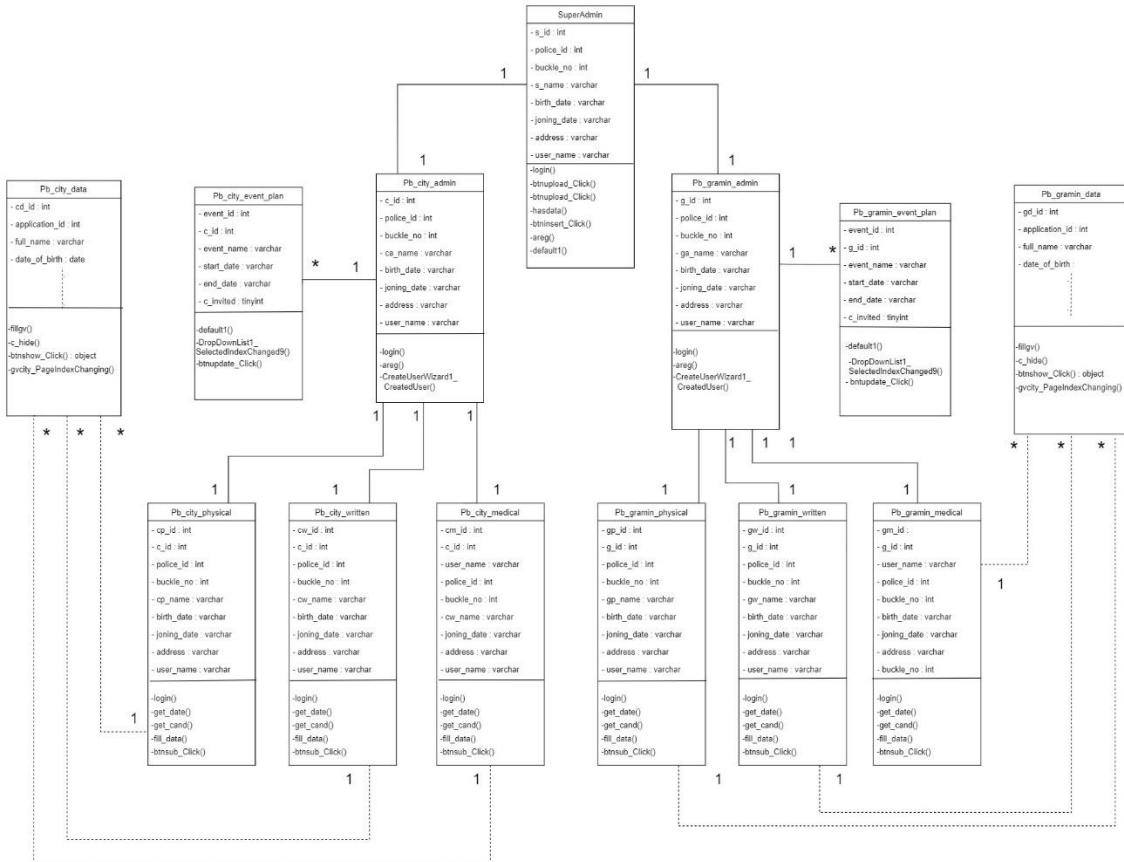
Gramin Physical:



Gramin Written:



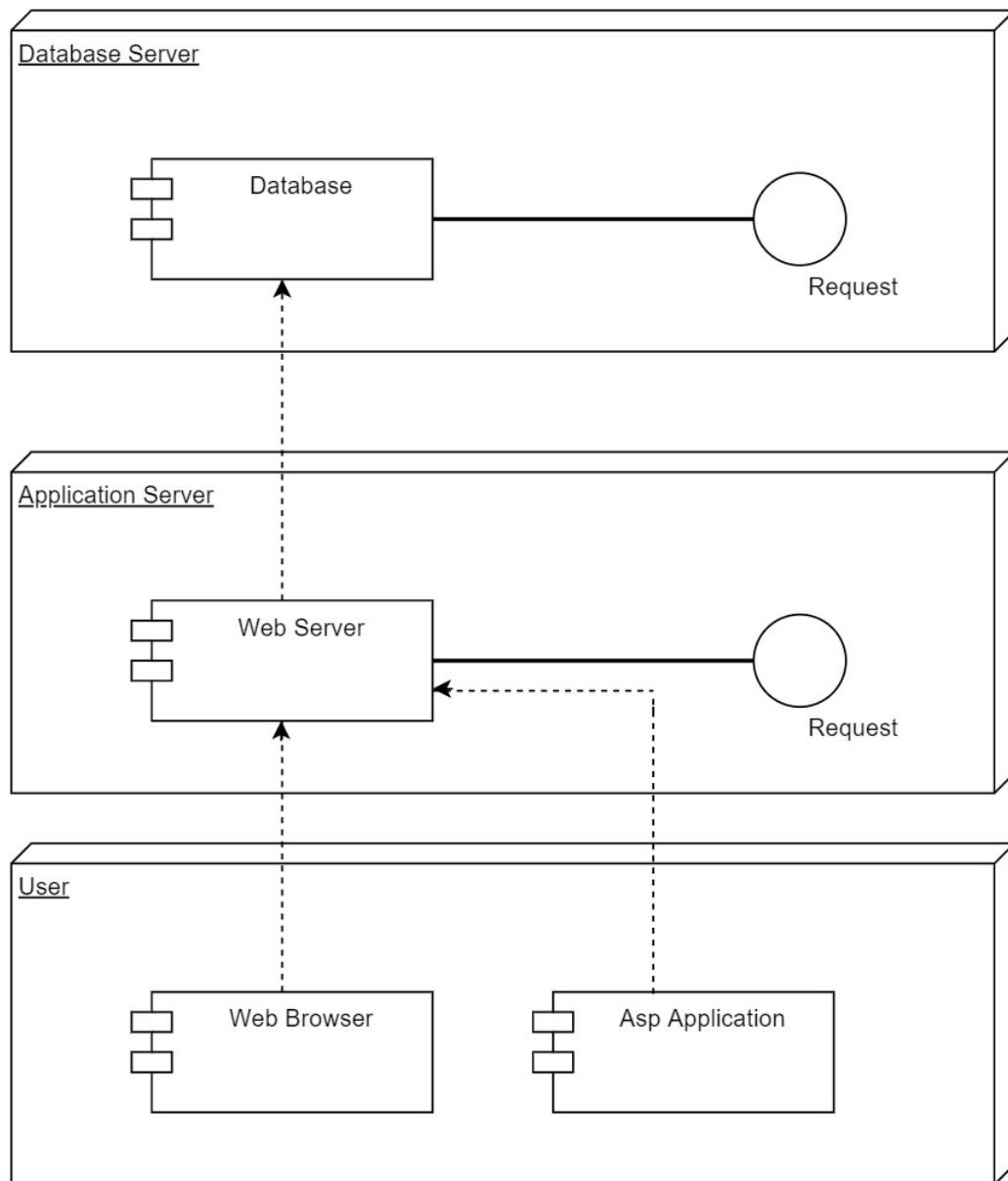
4.4 Class Diagram



4.5 Object Diagram:



4.6 Deployment Diagram:



4.8 User Interface

Login Page:

The screenshot shows a web browser window with the URL `localhost:50651/Sjlc0lhezzg5rukkywvll5kr/LoginPage`. The page title is "Log In". It contains two input fields: "Username" with placeholder text "Enter your Username" and "Password" with placeholder text "Enter your password". Below these fields is a checkbox labeled "Remember me next time." and a green "Log In" button.

City Admin Registration:

The screenshot shows a web browser window with the URL `localhost:50651/Sjlc0lhezzg5rukkywvll5kr/Admin/City_Admin.registration`. The page title is "City Admin Registration". On the left, there is a sidebar with icons for user management. The main content area has six input fields: "ID No.", "Buckle No.", "Name", "Birth date", "Joining date", and "Address". Below these fields is a "Next" button. At the bottom, there are links for "Admin Home." and "Version 3.1.0-rc".

Gramin Admin Registration:

Gramin Admin Registration

ID No:

Buckle No:

Name :

Birth date:

Joining date:

Address:

[Admin Home.](#) Version 3.1.0-rc

Add Records:

Import Excel File: No file selected.

Sheet Name:

Application Id	Full Name	Date Of Birth	District	Gender	Cast	Category	Email	Phone Number	Region
A000001	Nilesh Sakpal	24/09/2013	Pune	M	Open	Maratha	a@a.com	9999	City
A000002	Mahesh Raut	80/08/1999	Pune	M	OBC	Bhandari	m@m.com	9999	Gramin
A000003	Ramesh Lalit	24/09/2013	Pune	M	SC	Mahar	r12@gmail.com	9999	City
A000004	Payal Deshmukh	24/09/2013	Pune	F	SC	Mang	p12@gmail.com	9999	City
A000005	Saurabh Meheta	24/09/2013	Pune	M	SC	Mahar	s12@gmail.com	9999	City
A000006	Rajesh Purohit	24/09/2013	Pune	M	SC	Chambhar	rp12@gmail.com	9999	City
A000007	Prerna Shingh	24/09/2013	Pune	F	ST	Katkari	ps12@gmail.com	9999	Gramin
A000008	Keshav Pandit	24/09/2013	Pune	M	ST	Koli	kp12@gmail.com	9999	City
A000009	Vijender Shingh	24/09/2013	Pune	M	ST	Pardhi	vs@gmail.com	9999	City
A000010	Anjali Salvi	24/09/2014	Pune	F	ST	Koli	as123@gmail.com	9999	City

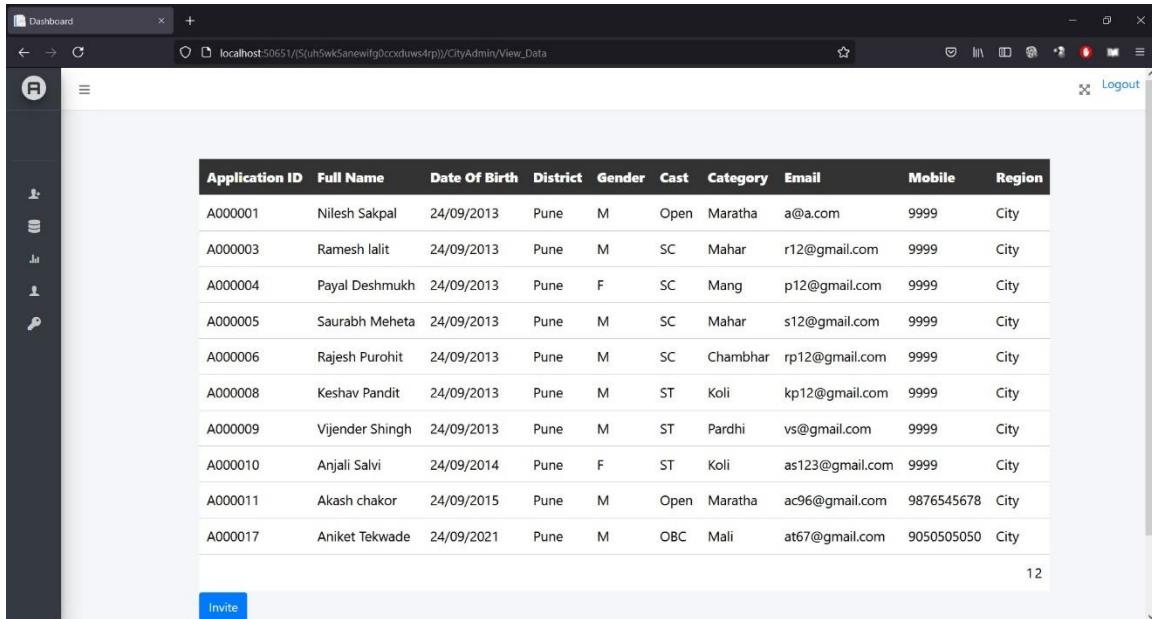
City Admin Homepage:

The screenshot shows the City Admin Homepage. On the left, a sidebar menu includes options like Home, Registration, Ground Test, Written Test, Medical Officer Registration, View Data, View Results, and Change-Password. The main area displays four cards: 'Total Candidates' (17, 04/01/2021 to 14/01/2021), 'Physical' (12, 04/01/2021 to 14/01/2021), 'Written' (5, 22/01/2021 to 24/01/2021), and 'Medical' (2, 22/01/2021 to 24/01/2021). Below these are gender breakdowns: Male (12) and Female (5). At the bottom, it says 'Admin Home.' and 'Version 3.1.0-rc'.

Ground Test Admin Registration:

The screenshot shows the 'Ground Test Admin Registration' form. It contains fields for ID No., Buckle No., Name, Birth date, Joining date, and Address. A 'Next' button is at the bottom right. The sidebar on the left shows registration-related icons. At the bottom, it says 'Admin Home.' and 'Version 3.1.0-rc'.

View Data:



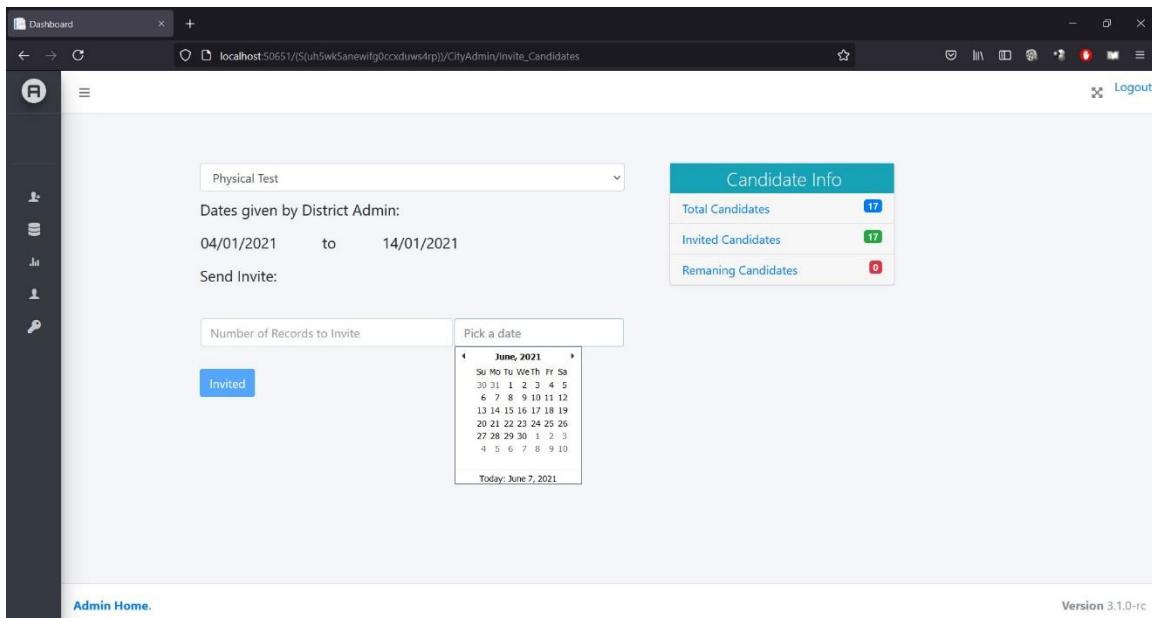
A screenshot of a web browser displaying a table of candidate data. The table has columns for Application ID, Full Name, Date Of Birth, District, Gender, Cast, Category, Email, Mobile, and Region. The data shows 12 rows of candidates from A000001 to A000017, each with unique details like name, gender, email, and mobile number.

Application ID	Full Name	Date Of Birth	District	Gender	Cast	Category	Email	Mobile	Region
A000001	Nilesh Sakpal	24/09/2013	Pune	M	Open	Maratha	a@a.com	9999	City
A000003	Ramesh lalit	24/09/2013	Pune	M	SC	Mahar	r12@gmail.com	9999	City
A000004	Payal Deshmukh	24/09/2013	Pune	F	SC	Mang	p12@gmail.com	9999	City
A000005	Saurabh Meheta	24/09/2013	Pune	M	SC	Mahar	s12@gmail.com	9999	City
A000006	Rajesh Purohit	24/09/2013	Pune	M	SC	Chambhar	rp12@gmail.com	9999	City
A000008	Keshav Pandit	24/09/2013	Pune	M	ST	Koli	kp12@gmail.com	9999	City
A000009	Vijender Shingh	24/09/2013	Pune	M	ST	Pardhi	vs@gmail.com	9999	City
A000010	Anjali Salvi	24/09/2014	Pune	F	ST	Koli	as123@gmail.com	9999	City
A000011	Aakash chakor	24/09/2015	Pune	M	Open	Maratha	ac96@gmail.com	9876545678	City
A000017	Aniket Tekwade	24/09/2021	Pune	M	OBC	Mali	at67@gmail.com	9050505050	City

12

[Invite](#)

Invite Candidates:



A screenshot of a web browser displaying an invitation form. The form includes a dropdown for 'Physical Test', a date range selector for 'Dates given by District Admin' (04/01/2021 to 14/01/2021), and a 'Send Invite:' button. On the right, there's a 'Candidate Info' section showing 'Total Candidates' (17), 'Invited Candidates' (17), and 'Remaining Candidates' (0). Below the form is a date picker for 'Number of Records to Invite' with a calendar showing June 2021. At the bottom, there are links for 'Admin Home.' and 'Version 3.1.0-rc'.

Physical Test

Dates given by District Admin:
04/01/2021 to 14/01/2021

Send Invite:

Number of Records to Invite

Pick a date

June, 2021

Su Mo Tu We Th Fr Sa
30 31 1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 1 2 3
4 5 6 7 8 9 10

Today: June 7, 2021

Admin Home. Version 3.1.0-rc

City Admin Physical Homepage:

The screenshot shows a web browser window for the "Physical test Homepage". The URL is `localhost:50651/(S(oswlkjzjra5ohd3oztou4gd))/City_Physical/City_Physical_HomePage`. The page title is "Physical test Homepage". On the left, there is a sidebar with navigation links: Home, Physical/Ground, View all data, Tests (Physical selected), Ground, View Results (Physical selected), Edit-Profile, and Change-Password. The main content area displays the following statistics: "Total Candidates" (17), "Male" (12), and "Female" (5). The date range is "04/01/2021 to 14/01/2021". At the bottom, it says "Admin Home." and "Version 3.1.0-rc".

Physical test:

The screenshot shows a web application interface for a police recruitment process. At the top, the title "Police Bharti" is displayed. Below it, the specific module "Physical Test" is open in a browser window.

The main form contains the following fields:

- Date: 14/01/2021
- Candidate: A000001
- Name: Nilesh Sakpal
- Date Of Birth: 24/09/2013
- Gender: M
- Category: Maratha
- Cast: Open
- Height: 180
- Chest: 79 | 84
- Weight: 90

Below the form, there are two buttons: "Pass" (blue) and "Absent" (red).

To the right of the main form, there is a sidebar titled "Candidate Info" which displays the following statistics:

- Total Candidates: 13
- Remaning Candidates: 4
- Done Candidates: 6
- Absent Candidates: 3

Further down, there are two sections with height criteria:

- Height Criteria**:
 - Male: > 165 cms
 - Female: > 155 cms
- Male Chest Criteria**:
 - Minimum: > 79 cms
 - Maximum: Minimum + 5 cms

Ground Test:

The screenshot shows the 'Ground Test' application interface. On the left, there's a sidebar with icons for Home, Candidates, Tests, and Help. The main area has fields for Date (14/01/2021) and Candidate (A000008). Below this, candidate details are listed: Name (Nilesh Sakpal), Date Of Birth (24/09/2013), Gender (M), Category (Maratha), Cast (Open), and various performance metrics for different events. To the right, a 'Candidate Info' panel displays statistics: Total Candidates (4), Done Candidates (2), Remaining Candidates (1), and Absent Candidates (1). A vertical scroll bar is visible on the right side of the main content area.

Event	Value
Male 1600m	4.5
Female 800m	10.1
Male 100m	10
Female 100m	9
Male Long Jump	20
Female Long Jump	10
Male Shot Put	20
Female Shot Put	10

Physical Result:

The screenshot shows the 'Result Physical' application interface. It features a sidebar with icons for Home, Candidates, Tests, and Help. The main content area displays a table of candidate data. At the bottom, there's a note about sending data to the admin and links for Admin Home and Version 3.1.0-rc.

Gender	Application ID	Full Name	Date Of Birth	Gender	Height	Min Chest	Max Chest	Weight
Male	A000001	Nilesh Sakpal	24/09/2013	M	180	79	84	90
Female	A000008	Keshav Pandit	24/09/2013	M	175	80	85	76

Don't Send Before : 14/01/2021

[Send Data to Admin](#)

[Admin Home](#) [Version 3.1.0-rc](#)

Ground result:

The screenshot shows a web browser window titled "Result Ground". The URL is "localhost:50651/S(cswkyjzxra5ohd3cztou4gd)/City_Physical/City_Physical_Ground_Data". The page displays a table of applicants with their details and physical performance scores. A note at the bottom left says "Don't Send Before : 14/01/2021". A button "Send Data to Admin" is visible on the right. The sidebar on the left has icons for Home, Medical, Tests, View Results, Re-Medical, Edit-Profile, and Change-Password. The footer shows "Admin Home." and "Version 3.1.0-rc".

Male	Female	Application ID	Full Name	Date Of Birth	Gender	800m	100m	Long Jump	Shot Put	Total
		A000018	Swati Jadhav	24/09/2022	F	0 0	0 0	0 0	0 0	0 0 0
		A000023	Harsha fara	24/09/2027	F	1 25	1 22	10 25	9 25	97

Send Data to Admin

Don't Send Before : 14/01/2021

Admin Home. Version 3.1.0-rc

Medical Homepage:

The screenshot shows a web browser window titled "City-Medical Home". The URL is "localhost:50651/S(dfronpb3dplpgtsut0nfgp)/CityMedical/City_Medical_Home". The page displays a summary of candidates: "17 Total Candidates" from "22/01/2021 to 24/01/2021". It also shows gender distribution: "Male 12" and "Female 5". The sidebar on the left includes "View all data", "Tests", "View Results", "Re-Medical", "Edit-Profile", and "Change-Password". The footer shows "Admin Home." and "Version 3.1.0-rc".

17 Total Candidates
22/01/2021 to 24/01/2021

Male 12 Female 5

View all data Tests View Results Re-Medical Edit-Profile Change-Password

Admin Home. Version 3.1.0-rc

Medical Test:

The screenshot shows the 'Medical Tests' application interface. At the top, there are dropdown menus for 'Date' (22/01/2021) and 'Candidate' (A000027). To the right, a 'Candidate Info' panel displays statistics: Total Candidates (6), Remaining Candidates (2), Done Candidates (1), and Absent Candidates (0). Below this, candidate details are listed: Name (Nilesh Sakpal), Date Of Birth (24/09/2013), Gender (M), Category (Maratha), Cast (Open), Height (180), Age (26), and Weight (65). Under the 'Medical Test' section, results for Eye Test, Ear Test, Nose Test, and Physical are shown, each with 'Pass' or 'Fail' options. The 'Eye Test' section includes 'Eye Sight and Squint' with radio buttons for Pass (selected) and Fail.

Written Test:

The screenshot shows the 'Written Test' application interface. At the top, there are dropdown menus for 'Date' (14/01/2021) and 'Candidate' (A000004). To the right, a 'Candidate Info' panel displays statistics: Total Candidates (13), Remaining Candidates (10), Done Candidates (1), and Absent Candidates (2). Below this, candidate details are listed: Name (Nilesh Sakpal), Date Of Birth (24/09/2013), Gender (M), Category (Maratha), Cast (Open), Marathi (5), English (20), Reasoning ability (20), GK & Current Affairs (20), and History & Geography (20).

Written Test Homepage:

The screenshot shows the Written Test homepage. At the top, there's a header bar with a 'Dashboard' button, a search bar, and a 'Login' link. Below the header is a sidebar with the following menu items: Home, Written, Search, View all data, Tests, View Results, Edit Records, Edit-Profile, and Change-Password. The main content area displays the following summary statistics:

Total Candidates	Written
17	22/01/2021 to 24/01/2021

Below this, there are two boxes: one for 'Male' (12) and one for 'Female' (5). At the bottom of the page, it says 'Admin Home' and 'Version 3.1.0-rc'.

Written result:

The screenshot shows the Written result page. The URL is 'localhost:50651/PoliceBharti/CityWritten/City_Written_test_Data'. The page features a sidebar with icons for Home, View all data, Tests, View Results, Edit Records, Edit-Profile, and Change-Password. The main content area contains a table with the following data:

Application ID	Full Name	Marathi Score	English Score	Reasoning Score	GK & Current Affairs	History & Geography
A000001	Nilesh Sakpal	5	20	20	20	20

Below the table, a message says 'Don't Send Before : 24/01/2021'. On the right side, there is a button labeled 'Send Data to Admin'. At the bottom, it says 'Admin Home' and 'Version 3.1.0-rc'.

Change Password:

Dashboard +

localhost:50651/500cn3ics2ulsizpy5akbg2o/CityWritten/Change_Password

☰ Login

Change Your Password

Password:

New Password:

Confirm New Password:

Change Password Cancel

Admin Home Version 3.1.0-rc

4.9 Table Specifications:

Table Name: - pb_super_admin					
Description: - holds information of the super admin					
Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	s_id	int	—	primary key, auto increment	Admin ID
2	police_id	int	—	—	Police ID of the police officer
3	buckle_no	int	—	—	Buckle NO. of the police officer
4	name	varchar	100	—	Name of the officer
5	birth_date	varchar	20	—	Date of birth
6	joining_date	varchar	20	—	Date of joining
7	address	varchar	100	—	Address of the officer
8	user_name	varchar	30	—	Username of Officer

Table Name: - pb_city_admin**Description:** - holds information of the City Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	c_id	int	—	primary key, auto increment	Admin ID
2	s_id	int		FK to s_id	s_id from pb_super_admin table
3	police_id	int	—	—	Police ID of the police officer
4	buckle_no	int	—	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_gramin_admin**Description:** - holds information of the City Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	g_id	int	—	primary key, auto increment	Admin ID
2	s_id	int		FK to s_id	s_id from pb_super_admin table
3	police_id	int	—	—	Police ID of the police officer
4	buckle_no	int	—	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_city_event_plan**Description:** - holds information of City Events.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	event_id	int	—	primary key, auto increment	Admin ID
2	c_id	int	—	FK to c_id	c_id from pb_city_admin table
3	event_name	varchar	45	—	Name of the event
4	start_date	varchar	20	—	Start date of the event
5	end_date	varchar	20	—	End date of event
6	c_invited	tinyint	—	—	Boolean if the candidates are invited

Table Name: - pb_city_physical**Description:** - holds information of the City Physical Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	cp_id	int	—	primary key, auto increment	Admin ID
2	c_id	int		FK to c_id	c_id from pb_city_admin table
3	police_id	int	—	—	Police ID of the police officer
4	buckle_no	int	—	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_city_written**Description:** - holds information of the City Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	cw_id	int	—	primary key, auto increment	Admin ID
2	c_id	int		FK to s_id	c_id from pb_city_admin table
3	police_id	int	—	—	Police ID of the police officer
4	buckle_no	int	—	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_city_medical**Description:** - holds information of the City Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	cw_id	int	—	primary key, auto increment	Admin ID
2	c_id	int		FK to s_id	c_id from pb_city_admin table
3	police_id	int	—	—	Police ID of the police officer
4	buckle_no	int	—	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_gramin_event_plan**Description:** - holds information of Gramin Events.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	event_id	int	-	primary key, auto increment	Admin ID
2	g_id	int	-	FK to c_id	g_id from pb_gramin_admin table
3	event_name	varchar	45	—	Police ID of the police officer
4	start_date	varchar	20	—	Buckle NO. of the police officer
5	end_date	varchar	20	—	Name of the officer
6	c_invited	tinyint	-	—	Boolean if the candidates are invited

Table Name: - pb_gramin_physical**Description:** - holds information of the Gramin Physical Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	gp_id	int	-	primary key, auto increment	Admin ID
2	g_id	int	-	FK to g_id	g_id from pb_gramin_admin table
3	police_id	int	-	—	Police ID of the police officer
4	buckle_no	int	-	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_gramin_written**Description:** - holds information of the Gramin Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	gw_id	int	-	primary key, auto increment	Admin ID
2	g_id	int	-	FK to g_id	g_id from pb_gramin_admin table
3	police_id	int	-	—	Police ID of the police officer
4	buckle_no	int	-	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_gramin_medical**Description:** - holds information of the City Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	gw_id	int	-	primary key, auto increment	Admin ID
2	g_id	int	-	FK to g_id	g_id from pb_gramin_admin table
3	police_id	int	-	—	Police ID of the police officer
4	buckle_no	int	-	—	Buckle NO. of the police officer
5	name	varchar	100	—	Name of the officer
6	birth_date	varchar	20	—	Date of birth
7	joining_date	varchar	20	—	Date of joining
8	address	varchar	100	—	Address of the officer
9	user_name	varchar	30	—	Username of officer

Table Name: - pb_city_data**Description:** - holds information of the City Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	cd_id	int	-	primary key, auto increment	Candidate ID
2	application_id	int	-	—	Application ID of the candidate
3	full_name	varchar	200	—	Full name of the candidate
4	date_of_birth	varchar	20	—	Birthdate of the candidate
5	district	varchar	50	—	District of the candidate
6	gender	varchar	10	—	Gender
7	cast	varchar	20	—	Cast of the candidate
8	category	varchar	20	—	Category of the candidate
9	email	varchar	45	—	Email ID of the candidate
10	phonenumber	bigint		—	Phone number of the candidate
11	region	varchar	45	—	Region of the candidate
12	show_data	tinyint	-	—	Boolean if the candidates data is shown to city admin
13	show_invite	tinyint	-	—	Boolean if the candidates are invited

14	physical_date	varchar	20	—	Date of physical event
15	physical_flag	tinyint	-	—	Boolean if the candidates are invited for physical event
16	written_date	varchar	20	—	Date of written event
17	written_flag	tinyint	-	—	Boolean if the candidates are invited for written event
18	medical_date	varchar	20	—	Date for medical event
19	medical_flag	tinyint	-	—	Boolean if the candidates are invited for medical
20	height	float	-	—	Height
21	minchest	float	-	—	Minimum chest
22	maxchest	float	-	—	Maximum chest (by holding breath)
23	weight	float	-	—	Weight
24	p_flag	tinyint	-	—	Boolean if physical test is conducted
25	race1600	float	-	—	Time of 1600m race
26	r16m	int	-	—	Score of 1600m race
27	race800	float	-	—	Time of 800m race
28	r8m	int	-	—	Score of 800m race
29	race100	float	-	—	Time of 100m race
30	r1m	int	-	—	Score of 100m race
31	longjump	float	-	—	Distance of Long Jump
31	ljm	int	-	—	Score of Long Jump
32	shotput	float	-	—	Distance of the shotput
33	spm	int	-	—	Shotput marks

34	pullups	int	-	—	Number of pullups
35	pum	int	-	—	Marks of pullups
36	g_flag	tinyint	-	—	Boolean if ground test is conducted
37	ground_total	int	-	—	Total marks of ground test
38	cpg_submit	tinyint	-	—	Boolean if data is submitted to City admin
39	meweight	float	-	—	Weight of the candidate
40	rweight	tinyint	-	—	Boolean for weight criteria
41	eyetest	tinyint	-	—	Boolean for eye test criteria
42	eartest	tinyint	-	—	Boolean for ear criteria
43	nosetest	tinyint	-	—	Boolean for nose test
44	phandi	tinyint	-	—	Boolean for handicap
45	knee	tinyint	-	—	Boolean for Knocking knees
46	pchest	tinyint	-	—	Boolean for Pigeon Chest
47	feet	tinyint	-	—	Boolean for flat foot
48	vveins	tinyint	-	—	Boolean for Varicose Veins
49	flimbs	tinyint	-	—	Boolean for fractured Limbs
50	hrigidus	tinyint	-	—	Boolean for hallux rigidus

51	skin	tinyint	-	—	Boolean for Skin diseases
52	heartbeat	tinyint	-	—	Boolean for heart beat test
53	fingures	tinyint	-	—	Boolean for fingure's deformity
54	gендertest	tinyint	-	—	Boolean for Gender testing
55	anal	tinyint	-	—	Boolean for anal testing
56	testiclag	tinyint	-	—	Boolean for testical growth
57	hydrocele	tinyint	-	—	Boolean for hydrocele
58	pvttest	tinyint	-	—	Boolean for penis/vagina test
59	aids	tinyint	-	—	Boolean for AIDS
60	piles	tinyint	-	—	Boolean for piles
61	m_flag	tinyint	-	—	Boolean if medical test is conducted
62	m_comment	varchar	350	—	Comment if candidate fails any medical test
63	mar_score	int	-	—	Score of written Marathi test
64	eng_score	int	-	—	Score of written English test
65	reso_score	int	-	—	Score of written Reasoning ability test
66	gk_score	int	-	—	Score of written General knowledge test

67	hist_score	int	-	—	Score of written History test
68	w_flag	tinyint	-	—	Boolean if written test is conducted

Table Name: - pb_gramin_data**Description:** - holds information of the City Written Admin.

Sr. No.	Field Name	Data Type	Size	Constraint	Description
1	cd_id	int	-	primary key, auto increment	Candidate ID
2	application_id	int	-	—	Application ID of the candidate
3	full_name	varchar	200	—	Full name of the candidate
4	date_of_birth	varchar	20	—	Birthdate of the candidate
5	district	varchar	50	—	District of the candidate
6	gender	varchar	10	—	Gender
7	cast	varchar	20	—	Cast of the candidate
8	category	varchar	20	—	Category of the candidate
9	email	varchar	45	—	Email ID of the candidate
10	phonenumber	bigint	-	—	Phone number of the candidate
11	region	varchar	45	—	Region of the candidate
12	show_data	tinyint	-	—	Boolean if the candidates data is shown to city admin
13	show_invite	tinyint	-	—	Boolean if the candidates are invited
14	physical_date	varchar	20	—	Date of physical event

15	physical_flag	tinyint	-	—	Boolean if the candidates are invited for physical event
16	written_date	varchar	20	—	Date of written event
17	written_flag	tinyint	-	—	Boolean if the candidates are invited for written event
18	medical_date	varchar	20	—	Date for medical event
19	medical_flag	tinyint	-	—	Boolean if the candidates are invited for medical
20	height	float	-	—	Height
21	minchest	float	-	—	Minimum chest
22	maxchest	float	-	—	Maximum chest (by holding breath)
23	weight	float	-	—	Weight
24	p_flag	tinyint	-	—	Boolean if physical test is conducted
25	race1600	float	-	—	Time of 1600m race
26	r16m	int	-	—	Score of 1600m race
27	race800	float	-	—	Time of 800m race
28	r8m	int	-	—	Score of 800m race
29	race100	float	-	—	Time of 100m race
30	r1m	int	-	—	Score of 100m race
31	longjump	float	-	—	Distance of Long Jump
31	ljm	int	-	—	Score of Long Jump
32	shotput	float	-	—	Distance of the shotput
33	spm	int	-	—	Shotput marks
34	pullups	int	-	—	Number of pullups

35	pum	int	-	—	Marks of pullups
36	g_flag	tinyint	-	—	Boolean if ground test is conducted
37	ground_total	int	-	—	Total marks of ground test
38	cpg_submit	tinyint	-	—	Boolean if data is submitted to City admin
39	meweight	float	-	—	Weight of the candidate
40	rweight	tinyint	-	—	Boolean for weight criteria
41	eyetest	tinyint	-	—	Boolean for eye test criteria
42	eartest	tinyint	-	—	Boolean for ear criteria
43	nosetest	tinyint	-	—	Boolean for nose test
44	phandi	tinyint	-	—	Boolean for handicap
45	knee	tinyint	-	—	Boolean for Knocking knees
46	pchest	tinyint	-	—	Boolean for Pigeon Chest
47	feet	tinyint	-	—	Boolean for flat foot
48	vveins	tinyint	-	—	Boolean for Varicose Veins
49	flimbs	tinyint	-	—	Boolean for fractured Limbs
50	hrigidus	tinyint	-	—	Boolean for hallux rigidus
51	skin	tinyint	-	—	Boolean for Skin diseases

52	heartbeat	tinyint	-	—	Boolean for heart beat test
53	fingures	tinyint	-	—	Boolean for fingure's deformity
54	gendertest	tinyint	-	—	Boolean for Gender testing
55	anal	tinyint	-	—	Boolean for anal testing
56	testiclag	tinyint	-	—	Boolean for testical growth
57	hydrocele	tinyint	-	—	Boolean for hydrocele
58	pvttest	tinyint	-	—	Boolean for penis/vagina test
59	aids	tinyint	-	—	Boolean for AIDS
60	piles	tinyint	-	—	Boolean for piles
61	m_flag	tinyint	-	—	Boolean if medical test is conducted
62	m_comment	varchar	350	—	Comment if candidate fails any medical test
63	mar_score	int	-	—	Score of written Marathi test
64	eng_score	int	-	—	Score of written English test
65	reso_score	int	-	—	Score of written Reasoning ability test
66	gk_score	int	-	—	Score of written General knowledge test
67	hist_score	int	-	—	Score of written History test

68	w_flag	tinyint	-	—	Boolean if written test is conducted
----	--------	---------	---	---	--------------------------------------

Chapter V

USER MANUAL

The User Manual describes the use of the system to Administrator. The user manual should be available as help.

5.1 User manual for District Admin:

- Firstly, district admin has to login to the system using the proper credentials.
- He can create Credentials for City and Gramin Admin.
- He has to upload excel file of candidates data to the system for further process.
- Verify the data and send it to the Gramin and City Admins respectively with tentative dates of the events.
- Managing existing Gramin and City admins like add new or remove previous admins.
- Changing his own credentials.
- Viewing the data of the candidates after the city and gramin admins conduct various tests.
- Generating reports based on the cast reservation.

5.2 User manual for City/Gramin Admin:

- City/Gramin admin has to login with the credentials provided by the District admin.
- Change his user credentials if necessary.
- Create three sub admins under him who supervise Physical, Written and Medical tests respectively
- Remove Physical, Written and Medical admins if necessary.
- View the data sent by the district admin and plan events in the given tentative dates.
- View the updated data of candidates with respective scores after physical, written and medical tests.
- Sent data to District admin for further evaluation.

5.3 User manual for Physical Admin:

- Physical admin has to login to the system by credentials provided by the Gramin/City admin
- Change the credentials if necessary
- View candidate data provided by the Gramin/City admin.
- Update the candidate's records with the physical (weight & chest) and ground tests score.
- View the scores of the candidates and update them with proper comment if necessary.
- View the reports of physical tests of the candidates.
- Send the data to Gramin/City Admin after conducting tests.

5.4 User manual for Written Admin:

- Written admin has to login to the system by credentials provided by the Gramin/City admin
- Change the credentials if necessary.
- View candidate data provided by the Gramin/City admin.
- Update the candidate's records with the written tests score.
- View the scores of the candidates and update them with proper comment if necessary.
- View the reports of written tests of the candidates.
- Send the data to Gramin/City Admin after conducting tests.

5.5 User manual for Medical Admin:

- Medical admin has to login to the system by credentials provided by the Gramin/City admin
 - Change the credentials if necessary
 - View candidate data provided by the Gramin/City admin.
-

- Update the candidate's records with the results of the medical tests conducted by the Doctor.
- Conduct the remedical of if candidates wish to appear again and update their test results.
- View the reports of medical tests of the candidates.
- Send the data to Gramin/City Admin after conducting tests

Chapter VI

BENEFIT, LIMITATIONS & CONCLUSION OF THE SYSTEM

6.1 Benefits of the system

- As this is a web based application, it can be used on any number of computers connected to the network.
- This application saves the paperwork used to store scores of the candidates.
- It can be easily tracked who updates a specific candidate score or details as it stores the UserID of the user.
- Allows for faster service.
- Easy, user friendly GUI.
- The system allows to add records in database rather than in regular old fashioned paperwork.
- The proposed system is extremely transparent.

6.2 Limitations of the System

- When SQL Server connection is not established the values from the database cannot be retrieved.
- System is designed for a single district at the moment.
- The user must be having authorization.
- Candidates cannot view their scores online.

6.3 Conclusion of the System

This was our project of System Design Lab about “Police Bharti”. Development of this System takes a lot of efforts from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this system. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fulfil to us.

Chapter VII

PROPOSED ENHANCEMENT



7.1 Multiple District Admins

Currently system has facility for just one District Admin, hence the system can only be used in one specific district. To use the system for whole state we need to allow permissions to create multiple District Admins in the system.

This will allow the system to be used on a larger scale and would benefit more users.

7.2 Allow candidates to view their scores

Currently the system is a closed Web Portal, hense only authorized persons have access to the information. We plan to create another domain and allow the candidates to access it so that they can view their scores online.

Candidate must only be allowed to view only his own scores, this can be achieved by asking him his candidate ID and sending an OTP on the registered mobile number of that candidate ID.

Chapter VIII

SELF-LEARNING FROM PROJECT

Project Based Learning is a teaching method in which we gain knowledge and skills by working for an extended period of time to investigate and respond to an authentic, engaging, and complex queries, problem, or challenge.

Self-directed learning about this process in which individuals take the initiative, with or without the help of others, in diagnosing our learning needs, formulating learning goals, identifying human and material resources for learning, choosing and implementing appropriate learning strategies, and evaluating learning.

Chapter VII

BIBLIOGRAPHY

Books:

- ASP.NET: The Complete Reference by Matthew Macdonald
- Build Your Own ASP.NET Website Using C# & VB.NET by Zak Ruvalcaba
- Introducing the MySQL 8 Document Store By Charles Bell

Websites:

- www.stackoverflow.com
- www.tutorialspoint.com
- www.c-sharpcorner.com
- dotnet.microsoft.com

ANNEXURE - III

SAMPLE PROJECT

CODE

Add Records.aspx:

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Admin/Admin_Master.Master"
AutoEventWireup="true" CodeBehind="Add_Records.aspx.cs"
Inherits="Police_Bharti.Admin.Add_Records" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">

    <script type = "text/javascript">

        function Filevalidation()
        {
            const fi = document.getElementById("<%=FileUpload1.ClientID%>");
            // Check if any file is selected.
            if (fi.files.length > 0) {
                for (const i = 0; i <= fi.files.length - 1; i++) {

                    const fsize = fi.files.item(i).size;
                    const file = Math.round((fsize / 1024));
                    // The size of the file.
                    if (file >= 1096)
                    {
                        alert("File too Big, please select a file less than 1mb");
                        var x = null;
                        document.getElementById("<%=FileUpload1.ClientID%>").value =
x;
                    }
                    else
                    {
                        document.getElementById('size').innerHTML = '<b>' +
                            file + '</b> KB';
                    }
                }
            }
        }

    </script>

    <table class="w-100">
        <tr>
            <td style="width: 191px"><h5>Import Excel File:</h5></td>
            <td>
                <asp:FileUpload CssClass="btn btn-sm" ID="FileUpload1" runat="server"
accept="application/vnd.ms-excel, application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet"/>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="FileUpload1" ErrorMessage="File Required"
style="color: #FF0000"></asp:RequiredFieldValidator>
            </td>
        </tr>
    </table>

```

```

        <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ControlToValidate="FileUpload1" ErrorMessage="Excel File required"
ValidationExpression="(.*)\.(xls|xlsx|XLS|XLSX)$" style="color:
#FF0000"></asp:RegularExpressionValidator>
        </td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td style="width: 191px"><h5>Sheet Name:</h5></td>
        <td>
            <asp:TextBox ID="sheettxt" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ControlToValidate="sheettxt" ErrorMessage="Sheet Name Required"
style="color: #FF0000"></asp:RequiredFieldValidator>
        </td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td style="width: 191px">
            <asp:Button CssClass="btn btn-primary btn-sm" ID="btnupload"
runat="server" Text="Upload" OnClick="btnupload_Click" />
        </td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td style="width: 191px">&nbsp;</td>
        <td class="text-center">
            <asp:GridView ID="gvExcelFile" runat="server" style="color: #000000;
font-size: large;" ForeColor="Black" AllowPaging="True" BackColor="White"
CellPadding="10" BorderColor="#CCCCCC" BorderWidth="1px" CellSpacing="2"
GridLines="Horizontal" OnPageIndexChanging="gvExcelFile_PageIndexChanging">
                <FooterStyle BackColor="#CCCC99" ForeColor="Black" />
                <HeaderStyle BackColor="#333333" Font-Bold="True"
ForeColor="White" />
                <PagerStyle BackColor="White" ForeColor="Black"
HorizontalAlign="Right" />
                <SelectedRowStyle BackColor="#CC3333" Font-Bold="True"
ForeColor="White" />
                <SortedAscendingCellStyle BackColor="#F7F7F7" />
                <SortedAscendingHeaderStyle BackColor="#4B4B4B" />
                <SortedDescendingCellStyle BackColor="#E5E5E5" />
                <SortedDescendingHeaderStyle BackColor="#242121" />
            </asp:GridView>
        </td>
        <td>&nbsp;</td>
</tr>
<tr>
        <td style="width: 191px; height: 80px;"></td>
        <td style="text-align:right; height: 80px;">
            <h6><asp:CheckBox ID="chkbx1" runat="server" Text="Delete previous
data" /></h6>
            <asp:Button CssClass="btn btn-success btn-sm" ID="btninsert"
runat="server" Text="Insert Records" OnClick="btninsert_Click"
CausesValidation="False" />
        </td>

```

```

        <td style="height: 80px"></td>
    </tr>
    <tr>
        <td style="width: 191px">&nbsp;</td>
        <td>
            <asp:Label ID="lblmsg" runat="server"></asp:Label>
        </td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td style="width: 191px">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>

</asp:Content>

```

Add Records.aspx.cs:

```

using System;
using System.Data;
using System.Data.OleDb;
using System.IO;
using System.Web.UI.WebControls;
using System.Drawing;
using System.Configuration;
using MySql.Data.MySqlClient;

namespace Police_Bharti.Admin
{
    public partial class Add_Records : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                btninsert.Visible = false;
                chkbx1.Visible = false;
                hasdata();
            }
        }

        protected void btnupload_Click(object sender, EventArgs e)
        {
            try
            {
                lblmsg.Text = "";
                string ConStr = "";
                string ext = Path.GetExtension(FileUpload1.FileName).ToLower();
                string path = Server.MapPath("~/Admin/Excel_Input/" +
FileUpload1.FileName);
                FileUpload1.SaveAs(path);
                if (ext.Trim() == ".xls")

```

```

        {
            ConStr = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
path + ";Extended Properties=\\"Excel 8.0;HDR=Yes;IMEX=2\\"";
        }
        else if (ext.Trim() == ".xlsx")
        {
            ConStr = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
path + ";Extended Properties=\\"Excel 12.0;HDR=Yes;IMEX=2\\"";
        }
        string query = "SELECT * FROM [" + sheettxt.Text + "$]";
        OleDbConnection conn = new OleDbConnection(ConStr);
        if (conn.State == ConnectionState.Closed)
        {
            conn.Open();
        }
        OleDbCommand cmd = new OleDbCommand(query, conn);
        OleDbDataAdapter da = new OleDbDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds, "a");
        gvExcelFile.DataSource = ds.Tables["a"].DefaultView;
        gvExcelFile.DataBind();
        ViewState["mydataset"] = ds;
        conn.Close();
        btninsert.Visible = true;
        chkbx1.Visible = true;

    }
    catch(Exception e1)
    {
        Response.Write("<script> alert('Error In Input'); </script>");
    }
}

protected void hasdata()
{
    if (gvExcelFile.Rows.Count != 0)
    {
        if (lblmsg.Text == "Records Overridden" || lblmsg.Text ==
"Successfully Added")
        {
            chkbx1.Visible = false;
            btninsert.Visible = false;
        }
        else
        {
            chkbx1.Visible = true;
            btninsert.Visible = true;
        }
    }
}

protected void btninsert_Click(object sender, EventArgs e)
{
}

```

```

        string connStr =
ConfigurationManager.ConnectionStrings["LocalMySqlServer"].ConnectionString;
        MySqlConnection con = new MySqlConnection(connStr);
        con.Open();
        MySqlCommand com;
        if (chkbx1.Checked)
        {
            com = new MySqlCommand("DELETE From pb_gramin_data", con);
            com.ExecuteNonQuery();
            com = new MySqlCommand("DELETE From pb_city_data", con);
            com.ExecuteNonQuery();

            int a = gvExcelFile.PageIndex;
            for (int i = 0; i < gvExcelFile.PageCount; i++)
            {
                gvExcelFile.SetPageIndex(i);
                foreach (GridViewRow g1 in gvExcelFile.Rows)
                {
                    if (((g1.Cells[9].Text) == ("city")) ||
((g1.Cells[9].Text) == ("City")))
                    {
                        com = new MySqlCommand("Insert into
pb_city_data(application_Id,full_name,date_of_birth,district,gender,cast,category,
email,phonenumer,region) values ('" + g1.Cells[0].Text + "','" + g1.Cells[1].Text
+ "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" + g1.Cells[4].Text
+ "','" + g1.Cells[5].Text + "','" + g1.Cells[6].Text + "','" + g1.Cells[7].Text +
"', '" + g1.Cells[8].Text + "','" + g1.Cells[9].Text + "')", con);
                        com.ExecuteNonQuery();
                    }
                    if (((g1.Cells[9].Text) == ("gramin")) ||
((g1.Cells[9].Text) == ("Gramin")))
                    {
                        com = new MySqlCommand("Insert into
pb_gramin_data(application_Id,full_name,date_of_birth,district,gender,cast,category,
email,phonenumer,region) values ('" + g1.Cells[0].Text + "','" +
g1.Cells[1].Text + "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" +
g1.Cells[4].Text + "','" + g1.Cells[5].Text + "','" + g1.Cells[6].Text + "','" +
g1.Cells[7].Text + "','" + g1.Cells[8].Text + "','" + g1.Cells[9].Text +
"', con");
                        com.ExecuteNonQuery();
                    }
                }
                gvExcelFile.SetPageIndex(a);

                lblmsg.ForeColor = Color.Green;
                lblmsg.Text = "Records Overridden";
            }
        }
        else
        {
            int a = gvExcelFile.PageIndex;
            for (int i = 0; i < gvExcelFile.PageCount; i++)
            {
                gvExcelFile.SetPageIndex(i);
                foreach (GridViewRow g1 in gvExcelFile.Rows)

```

```

        {
            if (((g1.Cells[9].Text) == ("city")) ||
((g1.Cells[9].Text) == ("City")))
            {
                com = new MySqlCommand("Insert into
pb_city_data(application_Id,full_name,date_of_birth,district,gender,cast,category,
email,phonenumer,region) values ('" + g1.Cells[0].Text + "','" + g1.Cells[1].Text
+ "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" + g1.Cells[4].Text
+ "','" + g1.Cells[5].Text + "','" + g1.Cells[6].Text + "','" + g1.Cells[7].Text +
"', '" + g1.Cells[8].Text + "','" + g1.Cells[9].Text + "')", con);
                com.ExecuteNonQuery();
            }
            if (((g1.Cells[8].Text) == ("district")) ||
((g1.Cells[8].Text) == ("District")))
            {
                com = new MySqlCommand("Insert into
pb_gramin_data(application_Id,full_name,date_of_birth,district,gender,cast,category,
email,phonenumer,region) values ('" + g1.Cells[0].Text + "','" + g1.Cells[1].Text
+ "','" + g1.Cells[2].Text + "','" + g1.Cells[3].Text + "','" + g1.Cells[4].Text
+ "','" + g1.Cells[5].Text + "','" + g1.Cells[6].Text + "','" + g1.Cells[7].Text +
"', '" + g1.Cells[8].Text + "','" + g1.Cells[9].Text + "')", con);
                com.ExecuteNonQuery();
            }
        }
        gvExcelFile.SetPageIndex(a);
        lblmsg.ForeColor = Color.Green;
        lblmsg.Text = "Successfully Added";
    }
    con.Close();
    hasdata();
}

protected void gvExcelFile_PageIndexChanged(object sender,
GridViewEventArgs e)
{
    DataSet mydataset = (DataSet)ViewState["mydataset"];
    DataTable myydatatable = mydataset.Tables["a"];
    gvExcelFile.DataSource = myydatatable;
    gvExcelFile.PageIndex = e.NewPageIndex;
    gvExcelFile.DataBind();
}
}

```